





Université de Lorraine
École Doctorale IAEM Lorraine
Département de Formation Doctorale en Informatique

Centre de Recherche en Architecture et Ingénierie
UMR n°3495 CNRS / Ministère de la Culture et de la Communication
École Nationale Supérieure d'Architecture de Nancy

---

Thèse

pour l'obtention du titre de

Docteur de l'Université de Lorraine

en Sciences de l'Architecture

par

**Daniel Zignale**

---

# Concevoir des services collaboratifs adaptés à des pratiques métiers : une méthode centrée usages

## Application au domaine de la construction

---

Soutenance publique le 17 juillet 2013

**Membres du Jury :**

| | | |
|---|---|---|
| Mme Sophie DUPUY-CHESSA | Rapporteur | Maître de conférences en informatique, HDR, Grenoble |
| M. Stéphane HANROT | Rapporteur | Architecte, Professeur en architecture, HDR, Marseille |
| M. Eric DUBOIS | Examinateur | Professeur en informatique, Namur |
| M. Pierre LECLERQ | Examinateur | Ingénieur-Architecte, Professeur en sciences appliquées, Liège |
| M. Gilles HALIN | Directeur | Maître de conférences en informatique, HDR, Université de Lorraine |
| M. Sylvain KUBICKI | Co-directeur | Architecte, Docteur en sciences de l'architecture, Luxembourg |

---

Avec le support du CRP Henri Tudor

et du Fonds National de la Recherche Luxembourgeois







*À Angelo et Tina…*

## Remerciements

Cette thèse est le fruit de l'engagement et la confiance de plusieurs personnes et instituts sans lesquels un jeune architecte fraîchement diplômé n'aurait pu mener un tel projet. Je tiens à leur adresser toute ma gratitude.

Je dois très probablement les fondations de ce projet à M. Sylvain Kubicki, qui a su piquer ma curiosité et éveiller mon intérêt pour la recherche, alors même que j'effectuais mon stage de Master 2. C'est grâce à sa motivation que j'ai eu la volonté et la possibilité d'entreprendre ce parcours qu'il a également su encadrer avec sérieux, mais aussi sympathie.

Je dois également le bon déroulement de cette étude à M. Gilles Halin, directeur de ce travail de thèse, et clé de voute de cette structure multidisciplinaire, entre architecture et informatique. Je lui suis reconnaissant pour sa pédagogie, la pertinence de ses propos, son sérieux et sa bonne humeur tout au long de la direction de mes travaux.

Je suis heureux d'avoir pu bénéficier du contexte professionnel offert par le CRP Henri Tudor et le CRAI. Je remercie Éric Dubois, directeur du département SSI, Séverine Mignon, coach de l'unité SISE, et Fabrice Absil, manageur du programme construction pour leur confiance et leur soutien. Je remercie également Jean-Claude Bignon, professeur à l'école d'architecture de Nancy et directeur de recherche au CRAI, pour ses conseils avisés.

Je remercie tout particulièrement mon collègue et ami Conrad Boton, pour sa bonne humeur, mais aussi son soutien et son aide précieuse. Les longues discussions qui ont régulièrement ponctué nos travaux de thèse respectifs ont toujours été très enrichissantes.

Nombreux sont mes collègues avec qui j'ai pu partager de très bons moments au cours de cette thèse et qui y ont contribué, même indirectement. Quelques noms parmi tant d'autres qui ont ma reconnaissance et ma sympathie : Jean-Sébastien, Thomas, Alain, Gerald, Cédric, Jocelyn, Elio, Carine, Kévin, Anis…

# Résumé

Dans le domaine du projet de conception-construction architecturale, la gestion de la collaboration entre les différents acteurs d'un projet est un enjeu important. D'un projet à un autre, en fonction du projet, mais aussi des acteurs qui y interviennent, les pratiques de travail varient. Parallèlement, de nombreux services sont proposés et utilisés pour assister la collaboration, certains sur un modèle « grand public » et d'autres plus orientés vers un usage professionnel. L'expérience CRTI-weB est un projet de développement d'un collecticiel mené avec et pour les professionnels du secteur au Luxembourg. Il propose actuellement deux services dans sa version commerciale : le service d'échanges et gestion de documents (plans, documents techniques) et le service de rédaction et gestion des rapports de chantier. Malgré l'étude des besoins métiers menée en amont du développement, l'outil présente cependant des manques d'adaptation. Ce constat d'inadaptation n'est pas un cas isolé, il reflète une lacune générale de ce genre d'outils à satisfaire pleinement les attentes des professionnels. C'est pourquoi ce travail doctoral propose un cadre d'analyse supporté par la modélisation des comportements des utilisateurs. Nous adoptons alors plusieurs points de vue relatifs à différents champs de recherches : le génie logiciel, la conception d'interfaces homme-machine, l'entreprise orientée services et la conception de système d'information, et enfin le travail collaboratif assisté par ordinateur (TCAO).

D'un point de vue organisationnel, nous identifions les pratiques collectives exercées par les groupes d'acteurs impliqués dans le projet. Nous en déduisons ensuite les pratiques individuelles, à savoir les responsabilités de chacun en fonction de son rôle dans le groupe.

D'un point de vue opérationnel, nous nous intéressons à la médiatisation de ces responsabilités par l'usage de différentes solutions technologiques. La caractérisation de ces usages est liée à la notion de contexte : le contexte technique (matériel, logiciel…), le contexte spatial (localisation, environnement…), le contexte temporel (fréquence, régularité, synchronisation…).

Enfin, le point de vue fonctionnel vise à définir le service utilisé, c'est-à-dire le comportement non plus de l'utilisateur, mais du système qui répond à son besoin.

Les concepts introduits et leur caractérisation permettent individuellement de cadrer et formaliser (par des modèles graphiques) la description des différents comportements à des fins analytiques. Dans un but productif, nous proposons la méthode PUSH (Practices and Usages based Services enHancement) qui orchestre ces différents points de vue et permet de générer un ensemble d'exigences pour le développement de services dits « adaptés ». Communication et traçabilité sont les maitres mots de cette méthode de conception.

Le contexte d'étude à la fois orienté recherche et développement (créé au travers de la collaboration entre le CRAI à Nancy et le CRP Henri Tudor à Luxembourg) nous a permis d'évaluer et d'améliorer la définition des concepts mis en avant ainsi que la mise en place de la méthode PUSH. L'amélioration des services du collecticiel CRTI-weB a fait l'objet de deux cas d'étude. Un troisième cas concernait le développement d'un tableau de bord de gestion de chantier sur interface mobile.

**Mots-clés :** Construction, Collaboration, Services informatiques, Génie Logiciel, Interfaces Homme-Machines, Conception centrée usages, Science de la conception.

# Sommaire

# Liste des abréviations et acronymes

| | |
|---|---|
| AIC : | Architecture Ingénierie et Construction |
| GL : | Génie Logiciel |
| GMF : | Graphical Modeling Framework |
| IDM : | Information Delivery Manual |
| IHM : | Interface Homme-Machine |
| MDA : | Model-Driven Architecture |
| MDE : | Model-Driven Engineering |
| MO : | Maitre d'ouvrage |
| MOE : | Maitre d'œuvre |
| MMPM : | Méta-Modèle des Pratiques Métiers |
| MMPU : | Méta-Modèle des Usages |
| MMS : | Méta-Modèle des Services |
| MMSA : | Méta-Modèle des Services Adaptés |
| MVC : | Model-View-Controller |
| PUSH : | Practice and Usage-based Service enhancement |
| SOA : | Service-Oriented Architecture |
| TCAO : | Travail Collaboratif Assisté par Ordinateur |

# Introduction

Au cours d'un projet de conception/construction architecturale, le contexte de travail, les compétences et les missions de chacun des intervenants sont différents. Cela rend difficiles la gestion et l'optimisation du travail et des échanges d'information dans un contexte collectif. Pourtant, à l'échelle du projet, l'objectif est unique : la création d'un ouvrage construit, unique et remarquable, qui sera parcouru, vécu, et adopté.

Les mauvais choix, les changements dans la conception, les retards, les malfaçons ont des impacts financiers considérables dans un contexte économique déjà particulier, l'erreur individuelle se répercutant sur le travail global. Il est donc de la responsabilité de chacun de mener à bien son activité. Mais souvent la manière dont les acteurs collaborent est une source de problèmes supplémentaires : mauvaise compréhension, mauvaise transmission d'informations, absence de suivi des demandes des collaborateurs, etc. ... La gestion de ces risques issus de la collaboration est un enjeu important.

Pour assister l'implication des acteurs dans leur collaboration avec les autres membres du projet, il est profitable d'utiliser les outils technologiques aujourd'hui disponibles. De plus en plus performants et accessibles, ces outils sont souvent la clé d'une bonne gestion du projet, diminuant les risques d'erreur et réduisant le temps de certaines activités de coordination. Les approches « orientées services » ambitionnent de rendre ces outils modulaires et flexibles. L'enjeu, auquel nous essayons de répondre au cours de ce travail, est alors d'assurer une bonne adéquation de ces services afin que la réponse aux besoins soit réelle.

Bien des architectes préfèrent que leurs graphistes aient eux-mêmes une formation d'architecte pour faciliter le dialogue entre eux. De même, profiter d'une connaissance accrue du métier pour spécifier des services peut être considéré comme un réel atout, bien qu'une équipe de conception doive s'entourer également des meilleures compétences techniques et ergonomiques. La caractéristique de ce travail de doctorat, fruit de la collaboration entre le Centre de Recherche en Architecture et Ingénierie de Nancy et le Centre de Recherche Public Henri Tudor à Luxembourg (et plus particulièrement son département « Service Science and Innovation »), est

justement de placer les préoccupations métiers au centre des projets de conception de services pour l'Architecture, l'Ingénierie et la Construction (AIC).

### Une thèse à la croisée des chemins entre sciences de l'architecture et génie logiciel

Les travaux universitaires menés en sciences de l'architecture couvrent plusieurs champs de recherche. Cette thèse en développera deux particuliers :

- la conception collaborative dans les projets d'Architecture, Ingénierie et Construction (AIC),
- la conception collaborative dans les projets de Génie Logiciel visant à assister les pratiques de projet AIC.

La conception est une science de l'artificiel, c'est un processus itératif de création et utilisation d'artefacts dont la nature est liée au domaine considéré. Par exemple, le plan ou l'ouvrage sont des artefacts de la conception architecturale alors que les modèles UML et le code implémenté comptent parmi ceux de la conception logicielle. Les processus quant à eux sont la mise en œuvre d'activités de différents types associées à des acteurs responsables de l'exécution de ces activités.

Les projets d'architecture et ceux du génie logiciel peuvent être appréhendés comme deux processus de conception similaires qui associent des points de vue différents. Ces points de vue peuvent, comme identifié plus haut, être source de divergences. Analyser ces projets c'est alors considérer plus que l'artificiel : nous verrons qu'un contexte de projet est aussi défini par des facteurs humains.

C'est à la croisée des chemins entre ces deux domaines que cette thèse trouve sa source. L'objectif est en effet de tirer parti d'une analyse des pratiques de conception dans le domaine AIC pour spécifier celles d'un projet de GL pour la conception de services adaptés au secteur. Cette problématique de double conceptualisation et de mise en correspondance sera au cœur de ce travail.

### Problématique générale

Notre analyse de la conception logicielle s'est particulièrement portée sur son aspect méthodique et structuré. En ce qui concerne l'analyse des pratiques architecturales, nous avons au contraire voulu garder à l'esprit le caractère unique et peu prédictif de chaque projet.

#### *Conceptualiser les méthodes de GL*

Les méthodes de conception logicielle ont évolué avec le temps pour répondre à la complexité croissante des applications informatiques, en s'orientant de la programmation proprement dite vers l'analyse et la modélisation. La première étape fut de pouvoir définir des parties de codes sous forme d'éléments réutilisables, facilitant ainsi la reprise de solutions informatiques prédéveloppées. Les étapes suivantes permirent de prendre de plus en plus de recul et de capitaliser les développements, cherchant à spécifier un système à partir de concepts externes relatifs aux utilisateurs, à leurs tâches, à leur métier, à leur organisation… Nous avons analysé et cherché à tirer parti de ces approches de conception dites « dirigées par les modèles ».

### *Conceptualiser les pratiques architecturales*

Malgré des bases fondamentales ancestrales, les pratiques de conception et de construction architecturales ne cessent d'évoluer. Elles sont poussées par l'innovation dans les techniques constructives et les matériaux, mais aussi par un contexte économique délicat qui demande d'être sans cesse plus productif et efficace. La volonté (et surtout la nécessité) d'atteindre des performances environnementales élevées se montre également particulièrement génératrice de changements. Une partie du présent travail de doctorat se tourne vers l'analyse et la description de ces pratiques. Ce travail met en avant les objectifs génériques et les spécificités d'un contexte de projet et ses variations.

### *Proposer des services informatiques pour le secteur AIC*

Comme le montre le schéma suivant (Figure 1), ce travail porte essentiellement sur l'expression des besoins (ou exigences) à l'origine d'un projet de conception de services, à partir de la description d'un projet de conception architecturale. En d'autres termes, nous proposons une approche qui intègre l'analyse des pratiques métiers dans le contexte AIC à une méthode de conception de services adaptés. Le concept d'usage médiatisant ces pratiques et matérialisé par les services est au centre de cette approche.

Cette méthode suit donc un processus en plusieurs étapes analytiques et conceptuelles. Ces étapes sont décrites par divers modèles qui formalisent l'expression de points de vue différents, relatifs aux différents intervenants dans un projet de développement de services (l'utilisateur, l'expert métier, le concepteur, le développeur). Le passage d'un modèle à l'autre définit l'évolution d'un point de vue vers un autre, jusqu'à la définition du service demandé. Cette évolution est définie par un méta-modèle qui conceptualise les points de vue et les correspondances entre chacun d'entre eux. Le plan de cette thèse découle des problématiques ici mises en avant.

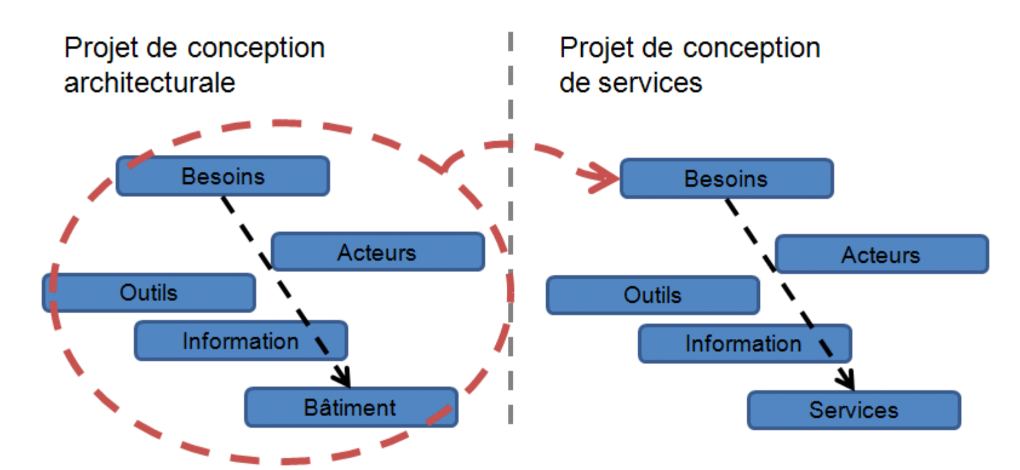



*Figure 1. Conceptualisation d'un projet de conception architecturale pour la spécification d'un projet de conception logicielle*

## Plan de la thèse

Trois parties composent ce manuscrit et intègrent les éléments de la problématique que nous venons de dégager. Elles décrivent une approche qui est elle-même une approche de conception : la conception d'une méthode pour l'adaptation de services au contexte collaboratif d'un projet AIC.

### *Partie 1 : assister la collaboration dans les projets de construction*

La première partie, composée de trois chapitres décrit les deux contextes de l'étude. Le premier est le contexte scientifique : une lignée d'études et notamment de thèses sur la coordination dans les projets de construction, son analyse, sa visualisation. Le deuxième est le contexte d'application : le développement de services d'assistance à la collaboration pour le secteur luxembourgeois de la construction. La problématique y est alors détaillée, suivie de la méthodologie adoptée, dans le chapitre trois.

### *Partie 2 : théories et méthodes, concevoir des services collaboratifs adaptés*

Quatre chapitres composent cette partie et explorent des champs d'études parallèles relatifs à la conception de services informatiques. Nous les présentons au travers de plusieurs paradigmes de la conception logicielle. Le premier chapitre (chapitre 4) décrit l'évolution des méthodes du GL, leur enrichissement par l'étude de l'utilisateur et introduit l'ingénierie dirigée par les modèles. Le deuxième (chapitre 5) explore le concept de service avec une étude sur la modélisation des processus métier dans une entreprise et la conception des systèmes d'information pour l'entreprise. Le troisième et le quatrième (chapitres 6 et 7) introduisent de manière théorique puis par des cas d'études la conception de services collaboratifs.

### *Partie 3 : guider la conception de services collaboratifs adaptés au secteur de la construction*

Les chapitres huit à douze forment la troisième et dernière partie de cette thèse. Le premier (chapitre 8) introduit notre approche pour l'adaptation de services collaboratifs aux pratiques du secteur AIC. Les suivants la décrivent selon le découpage suivant : modélisation des pratiques (chapitre 9), modélisation des usages (chapitre 10), modélisation des services et capitalisation de l'étude sous forme d'exigences (chapitre 11). Le dernier chapitre (chapitre 12) présente trois expérimentations qui ont été menées pour construire, parfaire et valider partiellement cette approche.

Une conclusion propose un bref récapitulatif du travail effectué. Nous essayons notamment d'en faire ressortir les limites et d'en dégager des perspectives qui pourront être abordées dans des travaux futurs.

# PARTIE 1 : Assister la collaboration dans les projets de construction. Définition d'une problématique et recul théorique.

Dans cette première partie, notre étude sera resituée dans son double contexte, entre enjeux métiers et démarche scientifique. Le chapitre 1 introduira le contexte métier, à savoir le domaine « Architecture, Ingénierie et Construction » (AIC), ses caractéristiques et les éléments qui le composent. Puis il approfondira plus particulièrement la thématique du Travail Collaboratif Assisté par Ordinateur (TCAO) dans ce contexte particulier. Le chapitre 2 décrira une expérience de conception de services d'un outil de TCAO particulier. Il nous permettra de tirer quelques conclusions quant à la nature et aux enjeux d'un tel travail de conception de services. À partir de ces analyses, nous définirons nos objectifs et notre problématique au cours du chapitre 3. Un premier recul théorique sur la science de la conception conclura ce chapitre et servira de point de départ à l'état de l'art proposé dans la deuxième partie de cette thèse.

# Chapitre 1 – La coordination dans les projets de conception/ construction architecturale

Ce premier chapitre fait le point sur les études menées autour de la coordination dans les projets de conception/construction architecturale, aussi appelés projets AIC (pour Architecture, Ingénierie et Construction). Il aborde un point de vue d'abord théorique visant à caractériser ce secteur sur la base des différents concepts que sont l'acteur, l'artefact et l'activité. Ces concepts sont issus de travaux précédents sur la modélisation de l'activité collective dans un projet AIC. Le chapitre introduit dans un second temps les outils de Travail Collaboratif Assisté par Ordinateur (TCAO) : leur buts, leur nature, leurs enjeux.

## 1.1 La caractérisation du secteur dans les travaux précédents

Les travaux de mes prédécesseurs, menés au sein du laboratoire MAP-CRAI, ont contribué à faire évoluer de manière continue une modélisation du contexte d'un projet AIC. L'objectif de Damien Hanser (Hanser 2003) était de représenter des situations collaboratives particulières au travers de trois concepts : l'acteur, l'activité et le document. Mouhamed Bouattour (Bouattour 2005) y ajouta le concept d'objet pour introduire la caractérisation des ouvrages (et éléments d'ouvrages) et des espaces bâtis. Sylvain Kubicki (Kubicki 2006) porta son attention sur le concept d'outil et la façon dont il s'intégrait dans le contexte du projet. L'objectif était alors d'utiliser la représentation de la collaboration afin  de mener une étude sur l'assistance de la collaboration grâce à des outils-logiciels dédiés. La thèse de Annie Guerriero (Guerriero 2009) considéra la représentation d'un aspect particulier de la collaboration, à savoir la confiance.

L'ensemble de ces travaux forme la base théorique sur laquelle s'appuie ce travail de thèse. Au cours des paragraphes suivants, nous présenterons les concepts importants qui en émergent et qui caractérisent ce qui est défini comme « le contexte de l'activité collective » d'un projet AIC.

### 1.1.1 *Les acteurs et leur caractérisation*

L'acteur est identifié comme une personne ou un groupe de personnes impliqué dans un projet. Les responsabilités, mais aussi la confiance qu'on accorde à un acteur dans un projet, sont portées par le rôle qu'il endosse. Selon (Hanser 2003), « les rôles sont le point fondamental de tout modèle destiné à représenter l'activité de groupe, car le rôle matérialise la participation d'un acteur à une activité » (p. 130). Il en identifie deux types : les rôles organisationnels et les rôles opérationnels.

Le rôle organisationnel est dépendant du projet et du cadre contractuel qui y est établi, il est donc défini en début de projet. Il peut être assimilé au rôle de l'acteur dans le domaine cinématographique, à savoir qu'un acteur peut jouer des rôles différents d'un projet à un autre (voire au sein d'un même projet). Les rôles les plus communément reconnus sont :

- Le maître d'ouvrage, qui est le commanditaire de l'ouvrage à concevoir
- Les architectes et ingénieurs, que l'on regroupera sous le rôle commun de concepteurs, responsable de la production des plans et autres documents qui spécifient le bâtiment à la fois formellement et techniquement
- Les constructeurs, qui mettent en œuvre le bâtiment à partir des documents de conception
- Le coordinateur, responsable du bon déroulement du projet, notamment par l'identification des problèmes et le suivi de l'information
- L'expert qui est relatif à un rôle de conseiller comme l'assistant à la maîtrise d'ouvrage, mais aussi d'analyste comme les bureaux de contrôle
- Enfin, nous regroupons toutes les institutions externes aux projets, mais ayant un rôle décisionnel sous le rôle d'administration

La notion de mission contractuelle définit les responsabilités attribuées à un acteur et ses objectifs à atteindre. Par exemple, les missions de maîtrise d'œuvre sont confiées par le maître d'ouvrage à un architecte ou un entrepreneur. En France, la loi MOP[1] (loi relative à la maîtrise d'ouvrage publique et à ses rapports avec la maîtrise d'œuvre privée) régit la nature de ces missions. Par définition, « le maître d'œuvre est la personne de droit privé ou le groupement de personnes de droit privé qui doit permettre d'apporter une réponse architecturale, technique et économique au programme »[2]. Les missions de maîtrise d'œuvre sont :

- les études d'esquisse,
- les études d'avant-projets,
- les études de projet,
- l'assistance apportée au maître de l'ouvrage pour la passation du contrat de travaux,
- les études d'exécution ou l'examen de la conformité au projet et le visa de celles qui ont été faites par l'entrepreneur,
- la direction de l'exécution du contrat de travaux,
- l'ordonnancement, le pilotage et la coordination du chantier,
- l'assistance apportée au maître de l'ouvrage lors des opérations de réception et pendant la période de garantie de parfait achèvement.

Le rôle opérationnel permet une distinction plus fine des acteurs dans le sens où il ne relève plus d'une responsabilité vis-à-vis du projet, mais vis-à-vis de l'information. Le rôle opérationnel conditionne, comme son nom l'indique, les opérations des acteurs sur les documents et sur l'ouvrage. Par exemple, en ce qui concerne le document « rapport de chantier » le rôle

[1] http://www.marche-public.fr/Marches-publics/Textes/Lois/loi-85-704-MOP.htm
[2] http://www.marche-public.fr/Marches-publics/Definitions/Entrees/Maître-oeuvre.htm

opérationnel de rédaction est attribué au coordinateur alors les autres acteurs n'auront qu'un rôle de consultation voire de commentaire.

D'un point de vue administratif, les acteurs en tant qu'individus et organisations sont identifiés par un titre. En ce qui concerne l'individu, ce titre dépend de sa formation, et correspond à son métier (ex. architecte, urbaniste, électricien, menuisier…). Pour une organisation on parlera de type d'organisation (ex. une agence d'architecture, un bureau d'études, une entreprise de gros œuvre…) (Hanser 2003).

Un acteur reconnu par son métier, peut endosser plusieurs rôles et donc missions. C'est souvent le cas pour un architecte qui peut être à la fois concepteur et coordinateur. Réciproquement, un rôle peut être joué par plusieurs personnes aux métiers différents. Par exemple, le même rôle de coordinateur joué dans un cas par l'architecte du projet peut aussi bien être attribué à un coordinateur pilote dédié qui fait de la coordination son activité principale. Grâce à une attribution des rôles ainsi plus étendue, les tâches sont plus finement réparties, ce qui permet à chacun de mieux concentrer son travail. Cela s'avère particulièrement nécessaire dans un projet de grande envergure, du fait de la complexité des missions.

De manière générale, le titre possède plus un caractère distinctif et témoigne de la formation antérieure d'un acteur. Le concept de rôle est beaucoup plus important, témoignant de l'activité réelle d'un acteur dans un contexte de projet particulier et du déroulement de celui-ci.

### *1.1.2 Les artefacts et les outils*

Une interaction entre un acteur et ce qu'il produit est supportée, instrumentée par des documents, des logiciels, des machines, des méthodes, des lois… : on dira qu'elle est médiatisée par des artefacts (Kuutti 1996; Engeström 1987; Hanser 2003). Comme le souligne (Hanser 2003), un artefact peut également être un produit. En effet, au cours d'un projet AIC, le document « plan » est dans un premier lieu un produit de conception puis est utilisé comme support lors de la construction de l'ouvrage.

Originellement donc, l'artefact est assimilé à un outil (les outils-machines, les outils-documents, les outils-méthodes, les outils-logiciels). (Kubicki 2006) propose cependant de traiter ces deux concepts distinctement :

- le concept d'outil hérite du caractère « instrumental » et contient aussi bien les outils-matériel que les outils-logiciel
- l'artefact est quant à lui ramené à sa définition première à savoir « un produit artificiel réalisé par l'homme » et regroupe les documents (les plans, les descriptifs textes, les photos…) et les objets (les ouvrages ou éléments d'ouvrages architecturaux).

### *1.1.3 L'activité*

De manière générale, l'activité est définie comme un ensemble « d'opérations humaines dirigées vers une finalité », « l'exercice d'une fonction, d'un métier»[3] ou « un ensemble de tâches

[3] http://www.linternaute.com/dictionnaire/fr/definition/activite/

organisées au sein d'un processus »[4]. L'activité peut également être synonyme d'action. Pourtant, ces différents termes se rapportent à des concepts sensiblement différents.

Nous pouvons trouver des premiers éléments de définition plus distinctifs dans la théorie de l'activité. Elle tire ses origines des travaux du psychologue Lev Vygotsky sur la « dynamique de l'activité humaine » et fut reprise par (Leontiev 1978) qui distingue trois concepts : l'activité, l'action, l'opération (voir (Hanser 2003)).

- L'**activité** est un processus de transformation visant atteindre un objectif général et découpé en plusieurs étapes ou phases.
- Les **actions** sont la décomposition d'une ou plusieurs activités en processus conscients et réfléchis. En d'autres termes, une action est la matérialisation d'une volonté d'atteindre un objectif (ou une partie d'objectif) au travers d'un processus dans lequel on évalue l'objectif, on exécute l'action et contrôle si l'objectif à été atteint.
- Les **opérations** sont les mécanismes qui composent l'action et qui sont réalisés de manière inconsciente.

Cependant, comme le soulève (Hanser 2003), il est difficile de cerner les limites entre ces trois concepts. Par exemple, selon ces définitions, le projet de construction d'un bâtiment est lui-même identifié comme activité. Il est cependant découpé en sous-activités relatives à des sous-objectifs comme réduire les couts, ou atteindre une performance thermique. De même, il est difficile d'identifier ce qui est de l'ordre de l'action ou de la sous-action au travers de tous les processus mis en œuvre dans un tel projet. Enfin, le caractère inconscient d'une opération est fortement dépendant de l'apprentissage de l'acteur qui l'exécute.

Sous le même concept d'activité, les travaux de nos prédécesseurs regroupent le projet, les phases de projet, et les tâches.

Le projet et les phases de projet

Il est admis de considérer un projet AIC comme un double processus composé d'une étape préparatoire (liée à la conception) et d'une étape opératoire (liée à la construction). Plus largement, ces deux étapes s'inscrivent dans un ensemble de **phases** du cycle de vie d'un bâtiment, dont la granularité varie selon les études (Bobroff et al. 1993; Armand 1997; Hanser 2003; Kubicki 2006). Nous retenons dans cette thèse les phases suivantes :

- le **montage** pendant lequel la maîtrise d'ouvrage acquiert le terrain, définit le programme du bâtiment à construire, établit le budget et choisit les concepteurs (éventuellement par le moyen d'un concours)
- la **conception** (aussi appelée phase « études ») pendant laquelle la maîtrise d'œuvre formule « une réponse architecturale, technique et économique » à la demande du maître d'ouvrage. Selon la loi MOP, elle comprend les (sous-)phases Esquisse, Avant Projet Sommaire (APS), Avant Projet Détaillé (APD), Projet et Assistance aux contrats de travaux.
- La **construction** (qui contient la mise en chantier et le chantier) pendant laquelle les différents corps de métiers réalisent les travaux et érigent le bâtiment conçu. Les constructeurs sont engagés par le moyen d'appels d'offres. Il est de la responsabilité du coordinateur de répartir les différentes interventions et de suivre le déroulement du chantier.
- L'**exploitation** qui fait suite à la réception du bâtiment et comprend la vérification, la gestion et la maintenance de celui-ci.

[4] http://fr.wikipedia.org/wiki/Activité

La fin de vie d'un bâtiment est souvent vue comme la dernière phase du projet. Nous préférons la considérer comme un nouveau projet en soi, avec sa propre étape préparatoire (supportée par l'étude de l'existant et des solutions à adopter) suivie d'une étape opératoire (la mise en œuvre des travaux de requalification ou sa démolition).

Le concept de tâche

Lorsqu'on parle de la répartition du travail dans un projet, aussi appelée **coordination** du projet, on utilise plus communément le terme tâche. Une tâche est une fraction d'activité attribuée à un membre du projet. Ce fractionnement diffère selon la nature de la répartition (Kvan 2000; Henri & Lundgren-Cayrol 2001; Kubicki 2006; Piquet 2009). (Malone & Crowston 1994) la définissent de manière générale comme « la gestion des dépendances entre les activités ».

- Dans le cas d'une coordination dite **collaborative**, le livrable d'une tâche est **une version** du produit final. Dans ce cas, chaque acteur participe pleinement à la résolution de l'objectif, c'est un travail « d'égal à égal ». Lors de la mise en commun, l'échange entre les différents acteurs est fort et le travail individuel devient difficilement identifiable. « Le travail collaboratif ne relève pas d'une répartition à priori des rôles » (Piquet 2009), il s'agit de la « résolution commune d'un problème » (Sylvain Kubicki 2006).
- Si la coordination est **coopérative**, chaque tâche donne lieu à **une partie** du produit final, ces parties étant ensuite assemblées. Cela nécessite une attribution précise de chaque tâche à un acteur en fonction de son rôle (voir 1.1.1). L'échange est faible et chacun est responsable de sa tâche, dans un groupe hiérarchiquement organisé. Les ressources sont également attribuées indépendamment pour chaque tâche.

Le passage d'une situation collaborative à une situation coopérative est d'ailleurs perceptible au cours des phases d'un projet AIC. La collaboration est relative aux activités conceptuelles, on l'observe par exemple en phase esquisse lorsqu'une équipe de concepteurs conçoit le bâtiment à partir du programme. Chacun en fait sa propre interprétation et propose sa version du bâtiment envisagé. La coopération est observable lors d'activités de formalisation ou d'exécution, comme le montage de l'APD par exemple. Les tâches comme le dessin des plans, des façades, des coupes, la réalisation des perspectives, etc…sont alors réparties individuellement selon les rôles de chacun. Le concepteur, alors supérieur hiérarchique supervise les tâches. Il en va de même sur le chantier, chacun étant responsable d'un élément de l'ouvrage à construire.

Qu'elles s'inscrivent dans une collaboration ou une coordination, les tâches peuvent être typées en fonction de leur nature et de ce qu'elles servent à accomplir. (Hanser 2003) identifie trois types :

- les tâches de production qui consistent à la manipulation d'objets en conception,
- les tâches de synthèse qui sont de l'ordre du décisionnel comme la validation ou le vote,
- les tâches de coordination qui sont de l'ordre de l'organisationnel, avec la gestion des ressources, des phases de travail, du personnel…

### *1.1.4 Conclusion*

Le schéma suivant (Figure 2), issu de (Guerriero 2009), retrace l'évolution des concepts que nous venons de décrire, composant le contexte coopératif dans les travaux précédemment menés au sein du laboratoire CRAI.

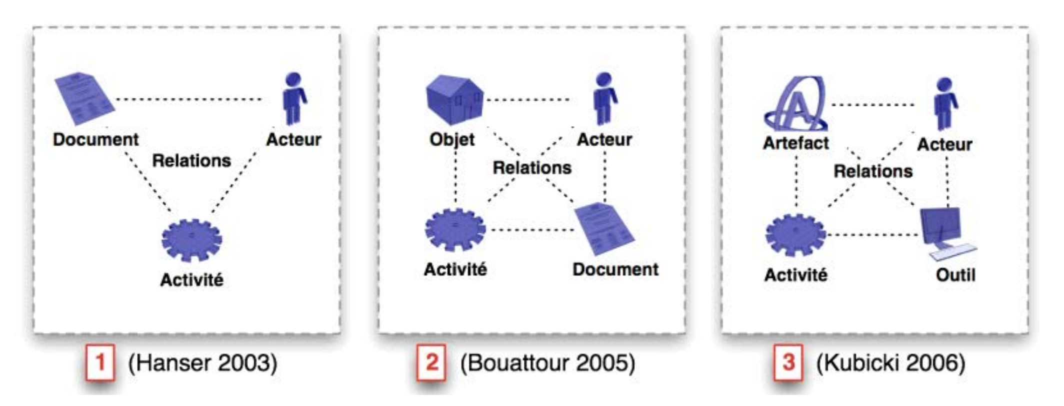



*Figure 2. Évolution des concepts du contexte de l'activité collective de 2003 à 2006 (issu de (Guerriero 2009))*

La dernière caractérisation du contexte coopératif est représentée par le diagramme relationnel qui suit (Figure 3).

Les relations entre chacun des concepts principaux (classes primaires) caractérisent ce que nous appellerons la dynamique du projet. Une relation entre deux éléments définit les actions ou les statuts d'un élément par rapport à un autre. Par exemple :

- la relation acteur-artefact est relative à l'édition de documents ou encore à l'élaboration d'un ouvrage et peut en définir la nature (ex. un acteur « rédige » un document),
- la relation activité-activité détermine l'ordonnancement des tâches (ex. une tâche « succède » à une autre tâche).

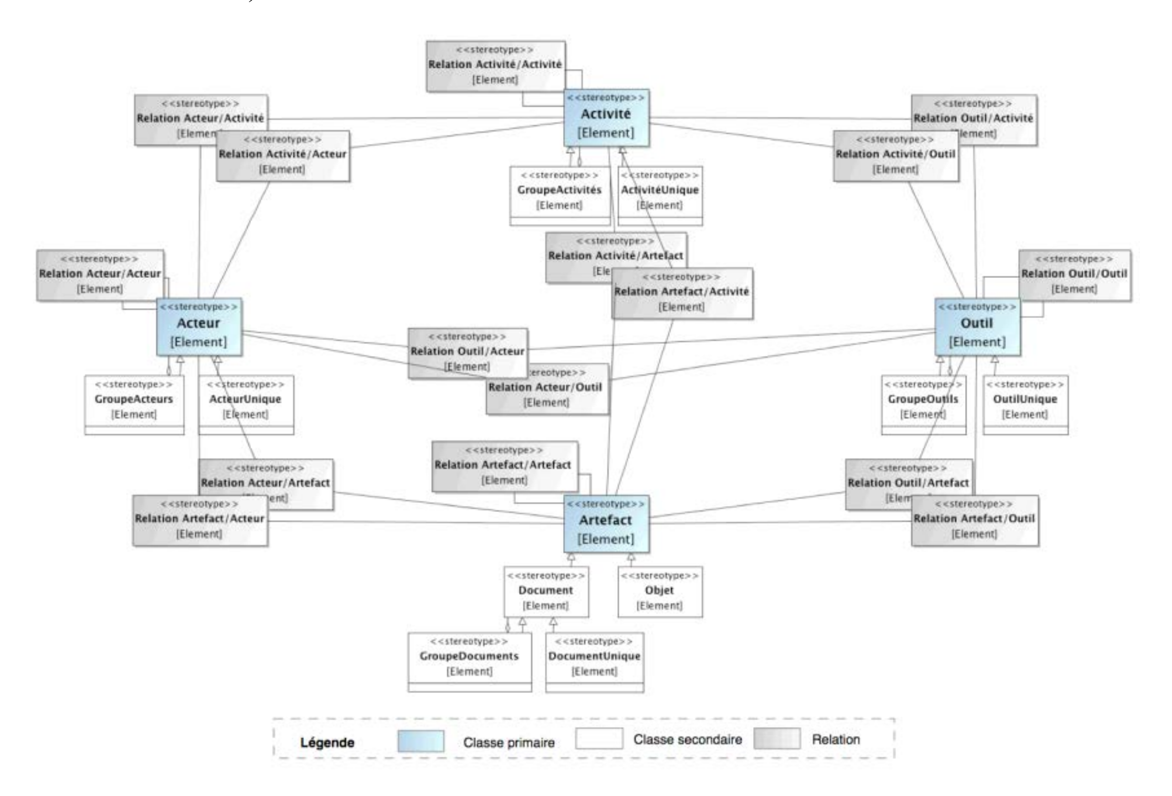



*Figure 3. Extrait du méta-modèle du contexte de l'activité collective (issu de (Guerriero 2009)*

Cependant, ces relations ne sont pas clairement définies. De plus, la description d'une situation collaborative de par ces relations reste ambiguës. Par exemple, si un document est créé par un acteur, il est aussi le fruit d'une activité et également produit par un outil. Il faudrait pouvoir définir clairement la nature de ces relations. Nous constatons donc un besoin de restructurer ce modèle.

Rappelons également que la caractérisation du contexte de l'activité collective a jusqu'à présent été menée afin de visualiser et comprendre de celui-ci. On parle d'ailleurs de multi-visualisation, le but étant de lier entre elles des représentations différentes comme le font par exemple les outils Bat'iViews (Kubicki 2006) et Bat'iTrust (Guerriero 2009).

Nous montrons dans ce travail que cette représentation peut aussi servir une approche plus productive en devenant la base de la conception d'outils.

On note à ce propos que dans sa thèse, Annie Guerriero ne considère pas l'outil lorsqu'elle traite de la confiance vis-à-vis du contexte d'un projet. Ce concept suscite en effet l'interrogation : l'outil fait-il partie du contexte d'un projet ou est-ce qu'il l'instrumente ? Doit-il être considéré en même de temps ou a posteriori des caractérisations du projet ?

Nous choisissons de considérer l'outil comme un moyen de médiatiser le contexte de l'activité collective d'un projet et non pas comme un élément de celui-ci. Cela nous permet en effet de traiter leur description de manière séparée (d'abord le métier puis l'outillage), cherchant ensuite à caractériser le lien qui s'opère entre activité collective et outils à travers cette médiatisation.

Le sous-chapitre suivant justifie le besoin d'une telle médiatisation et présente ensuite des exemples d'outils.

## 1.2 Les projets de construction et les outils de TCAO

### *1.2.1 Pourquoi médiatiser l'activité collective ?*

(Kubicki et al. 2006) relève l'importance de la maîtrise des processus de coordination pour assurer le succès d'un projet AIC et par extension, la qualité du bâtiment construit. Certains facteurs liés à la nature du projet AIC et porteurs de risques (c.à.d. susceptibles de s'opposer à son bon déroulement) font émerger l'importance de cette maîtrise :

- de nombreuses contraintes fonctionnelles, techniques, économiques et esthétiques qui varient d'un projet à un autre,
- de nombreux acteurs avec leurs propres connaissances et méthodes de travail, certains parfois réfractaires à adapter celles-ci à un projet particulier,
- des équipes éphémères qui se recomposent tout au long du projet,
- des relations contractuelles non hiérarchiques et des décisions non centralisées (particulièrement en phase chantier),
- un séquençage du projet basé sur des prises de décision « sur le tas », mais pourtant déterminantes et parfois irréversibles,
- des interactions entre acteurs informelles, peu tracées et peu prédictives.

Nous avons précédemment distingué coordination collaborative et coordination coopérative. Pour rappel, on peut dire que lors d'une coopération « on sait précisément ce qu'on va faire » alors que dans la collaboration, « on a l'idée de ce vers quoi on va » sachant que cela peut évoluer (Rameau & Samyn 2006). Afin de mieux comprendre la nature des processus de coordination, (Kubicki 2006) opère trois autres distinctions :

- entre coordination explicite (basée sur les artefacts) et implicite (orale),
- entre coordination hiérarchique (multi-acteurs), adhocratique (inter-acteurs) et transversale (extra-acteurs),
- entre ajustement mutuel, supervision et standardisation

La coordination explicite se base sur une ligne de progression bien établie et la résolution des problèmes par anticipation. Cependant, l'anticipation des problèmes n'est pas toujours possible, et il est nécessaire d'adapter le processus aux changements : on parle alors de coordination implicite (Andersen et al. 2000; C Godart et al. 2001).

La coordination hiérarchique ou multi-acteurs repose sur le partage de l'information pour le groupe et la conscience de celui-ci en ce qui concerne le déroulement du projet. La coordination adhocratique ou inter-acteurs caractérise un travail d'acteur à acteur au sein du projet alors que la coordination transversale ou extra-acteurs met en relation un acteur du projet avec un acteur dit « externe » (comme un expert ou un sous-traitant spécialisé). Dans les deux cas (inter/extra) on observe plutôt un phénomène de mise en commun de l'information basée sur l'échange, différent de la diffusion à sens unique dont les droits reviennent à une entité hiérarchique (Figure 4).




*Figure 4. Distinction entre coordination hiérarchique, adhocratique et transversale (tiré de (Kubicki 2006))*

Enfin à partir des travaux de Mintzberg (Mintzberg 1989), on distingue les mécanismes de coordination d'ajustement mutuel (basés sur la communication informelle), de la supervision (basée sur des ordres et des instructions) et de la standardisation (c.à.d. de la spécification des procédés, des résultats à obtenir ou encore des qualifications à avoir).

Le schéma suivant (Figure 5) illustre la corrélation entre les caractères de la coordination dans un projet AIC. Les activités impliquant un grand nombre d'acteurs, typiquement les activités opératoires comme celles de construction en phase chantier, sont en général de nature coopérative et nécessite d'être organisées de manière explicite. Les activités préparatoires, relatives à la conception en collaboration, reposent quant à elles sur la prise récurrente de décisions et l'auto-organisation du travail pondérée par des ajustements mutuels. Nous constatons également à travers ce schéma que les artefacts manipulés (les plans en sont le meilleur exemple) sont en constante évolution lors des activités préparatoires (artefacts dynamiques) alors qu'ils sont finalisés lors des activités opératoires (artefacts statiques).

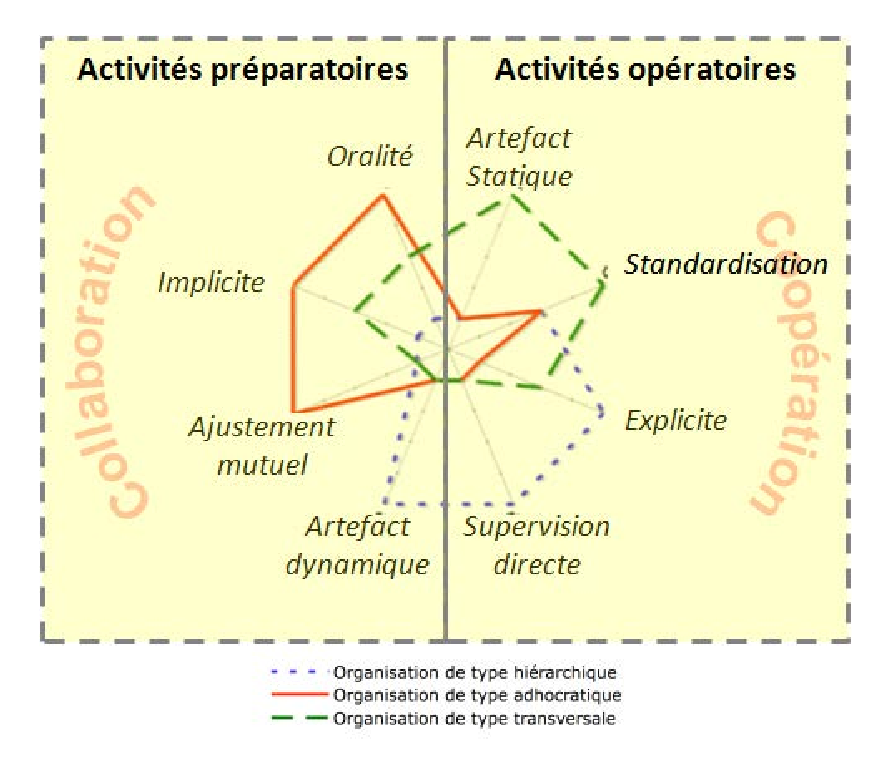



*Figure 5. Caractérisation des activités de coordination dans un projet AIC (adapté de (Kubicki 2006))*

La volonté de maîtriser la coordination et de faciliter l'auto-coordination a motivé l'utilisation d'outils médiatisant celle-ci. Pour cela il a fallu améliorer notre compréhension de cette activité collective et définir celle-ci sous toutes ses formes. Parallèlement, l'innovation constante en matière de technologies ouvre le panel des possibles, et favorise la création de nouveaux outils. Nous en présentons quelques types dans la section suivante, certains fréquemment utilisés et intégrés aux projets, d'autres plus innovants et prospectifs.

### *1.2.2 Les types d'outils*

De manière générale, les outils supportant la coordination permettent d'étendre les fonctionnalités de nos outils classiquement utilisés pour gérer nos activités, dans un cadre pluri-utilisateur. L'exemple le plus courant est le calendrier que nous pouvons utiliser seul ou a plusieurs. Les outils dont nous parlerons sont des outils numériques et non pas des outils dits « physiques » (comme l'engin de chantier).

Médiatiser la coordination dans les activités opératoires

Les activités considérées ici sont relatives à la construction du bâtiment et plus particulièrement à sa gestion, son organisation, son contrôle. Plusieurs types d'outils sont généralement adoptés pour médiatiser ces activités opératoires (Guerriero 2009).

- Les outils de planification permettant de coordonner les différentes tâches dans le temps. Il existe plusieurs méthodes de planification comme les diagrammes de Gantt (exemple Figure 6), PERT ou « Line Of Balance » (Yamin & Harmelink 2001).

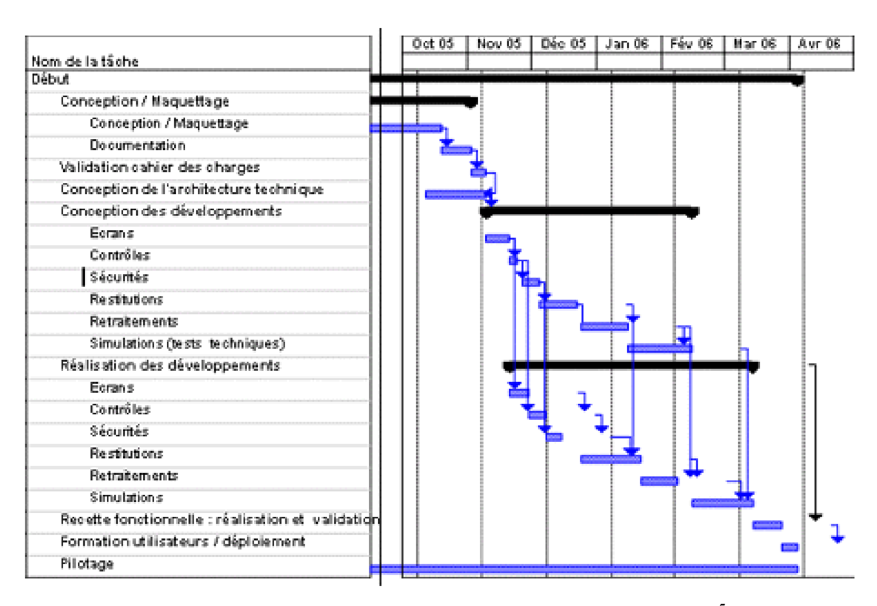



*Figure 6. Exemple de diagramme de GANTT* [5]

- Les outils de compte-rendu permettent de rédiger les comptes-rendus de réunion et autres rapports relatifs au chantier. Les outils de traitement de texte sont les plus couramment utilisés, mais le développement d'outils numériques dédiés « afin de structurer le contenu d'un document est apparu comme une perspective intéressante pour faciliter, tant l'édition, que la diffusion et la consultation du compte-rendu » (Guerriero 2009).
- Les outils de gestion des coûts et des ressources souvent sous la forme de tableurs ont pour but d'anticiper le budget lors des prises de décisions, mais aussi de gérer les paiements durant la construction de l'ouvrage.
- Les plateformes d'échange de documents permettent de centraliser les documents dans un espace partagé et de gérer l'accès des différents acteurs. Ils possèdent des fonctionnalités complémentaires telles que l'ajout de commentaires sur un document partagé, la notification par mail lors d'un partage…
- Les outils de mesure permettent d'évaluer les performances du bâtiment (ex. les performances énergétiques) sur la base du recueil de données, de la simulation et la génération d'une évaluation (relative ou absolue).
- Les outils de maquettage numérique sont une évolution des outils de Conception Assistée par Ordinateur (CAO) classiques. Ils permettant d'ajouter à un modèle 3D des informations relatives aux éléments de l'ouvrage tel que la composition, les propriétés, la technique de montage à employer…
- Les divers outils comme le téléphone, le mail, la messagerie instantanée… supportent la communication.

On parle d'outil 4D puis nD lorsqu'il s'agit de représenter un modèle 3D (ou une maquette numérique) et de lier les ouvrages à des tâches dans une vue planning qui apparait alors comme une quatrième dimension puis à d'autres informations comme la représentation des coûts, etc.…

Le tableau de bord est un outil particulier que l'on peut qualifier d'outil d'agrégation et de synthèse vis-à-vis des autres outils que nous venons de présenter. Il apparait comme un « instrument de mesure » (Fernandez 2010) en fournissant des valeurs synthétiques relatives à l'information traitée (ex. le statut d'une tâche, le nombre de remarques dans un rapport). Dans un

[5] Image tirée de http://www.gestiondeprojet.net/articles/gantt.html

second temps, il peut suggérer des compositions de vues contenant l'information générée par les autres outils et faisant état du déroulement du projet.

La médiatisation de la coordination dans les activités préparatoires

Comme nous l'avons défini, les activités préparatoires sont relatives à la conception et sont essentiellement basées sur la « mise en commun » d'information. Le schéma suivant (Figure 7) issu de (Hanser 2003) illustre cela par les phénomènes de conception distribuée et de co-conception.

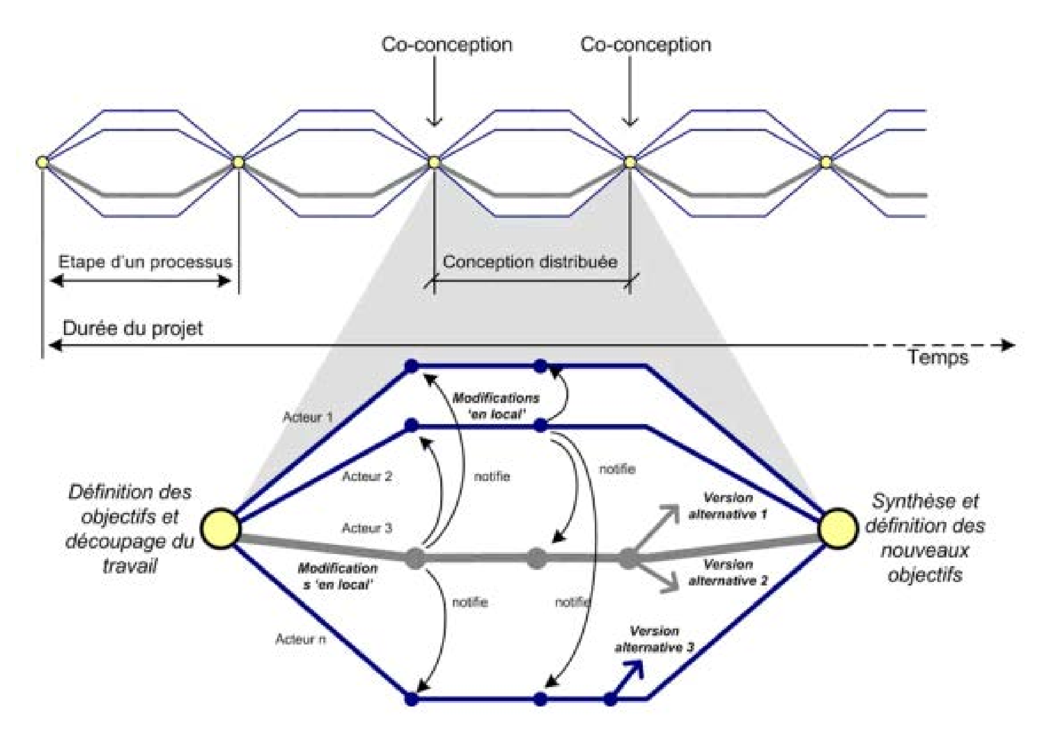



*Figure 7. Processus de conception alternant conception distribuée et points de synthèse (tiré de (Hanser 2003))*

Malgré l'ajustement mutuel qui caractérise la conception, la traçabilité des choix et des activités reste un atout et un besoin, notamment lors de la mise en commun et la synthèse des propositions. Les outils qui assistent la coordination en phase chantier répondent également aux besoins de la conception.

- Les outils de calendrier partagé permettent de fixer des dates limites, des rendez-vous, d'organiser des évènements et notamment planifier les points de synthèse…
- Les outils de partage permettent de diffuser des documents. L'intégration dans les outils de production permet l'édition par plusieurs acteurs. On trouve ce genre de fonctionnalités pour l'édition de textes (ex. avec Google Documents) mais aussi plus récemment pour la création de plans (ex. Autocad, voir Figure 8).
- La maquette numérique peut aussi être utilisée en situation de conception, l'ingénieur pouvant par exemple « récupérer le modèle établi par l'architecte en vue de dimensionner les structures ou encore de réaliser des simulations » (Guerriero 2009).

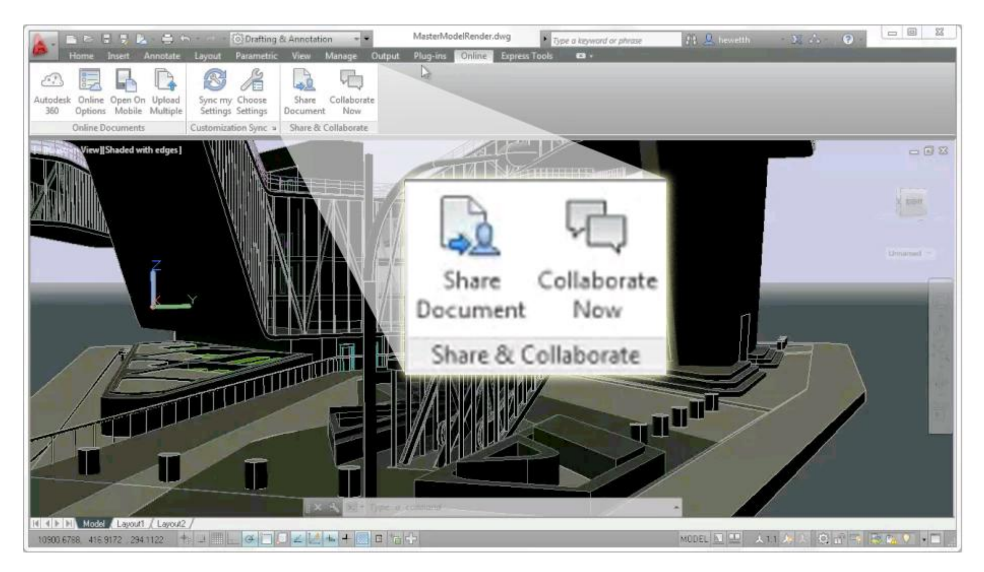



*Figure 8. Fonctionnalités collaboratives dans l'outil de CAO Autocad*

L'association de deux outils que sont le bureau virtuel (une table à dessin numérique couplée à un système de visioconférence) et le logiciel Sketsha (un logiciel de croquis distribués) (Figure 9) est un exemple de dispositif de médiatisation de la conception collaborative par l'esquisse. Il permet à « deux équipes distantes de collaborer dans des conditions reproduisant la coprésence […] de collaborateurs géographiquement dispersés ». Il est notamment utilisé dans un projet de co-conception en milieu universitaire nommé le Studio Digital Coopératif (SDC) (Safin et al. 2012).

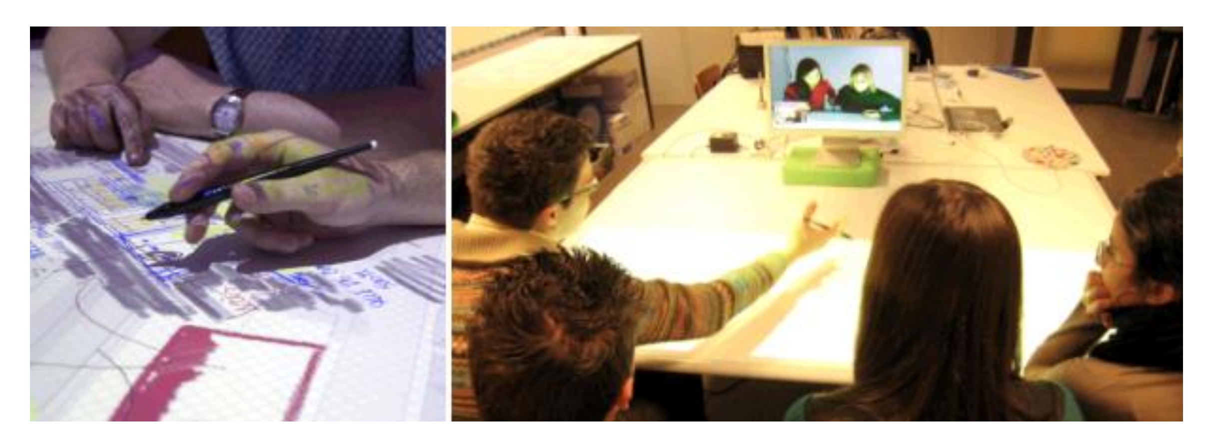


*Figure 9. Utilisation du dispositif « Bureau Virtuel + Sketsha » au cours du projet SDC (photos tirées de (Saffin & Leclercq 2010))*

### *1.2.3 L'adaptation au contexte*

Les outils proposés pour assister les activités des acteurs d'un projet AIC sont développés de manière à être « adaptés » au secteur. L'adaptation d'un outil est la manière dont celui-ci répond aux besoins de ses utilisateurs. Nous verrons que ces besoins sont étroitement liés à la notion de contexte.

(Kubicki 2006) présente le concept de contexte « tri-facettes » composé des trois contextes définissant le rapport d'un acteur à son activité au travers des outils qu'il manipule. Ces trois contextes sont (Figure 10) :

- le **contexte acteur**, qui caractérise les intentions et les préférences de l'acteur qui appréhende l'outil,
- le **contexte utilisateur**, qui caractérise l'interaction avec l'outil, l'outil lui-même mais aussi son environnement,
- le contexte de l'activité collective, caractérisant le projet.

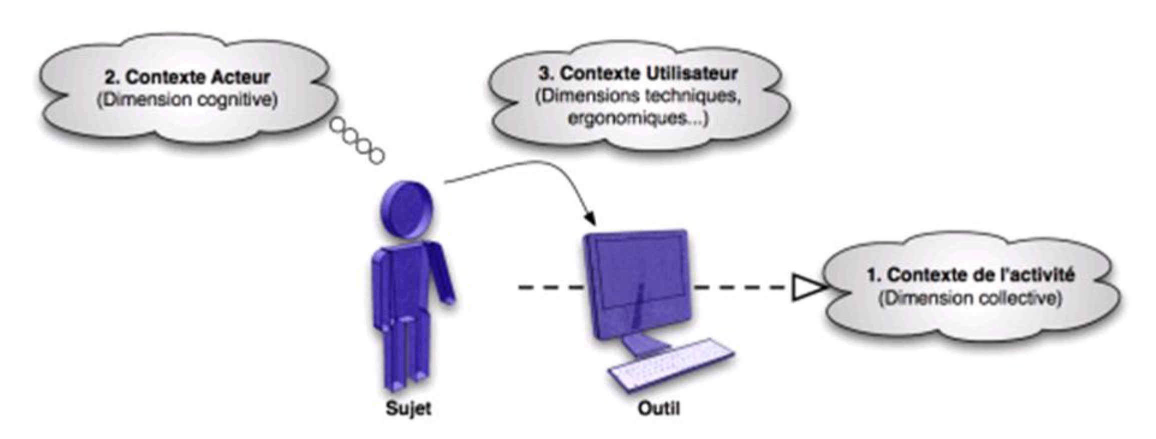



*Figure 10. Le contexte « tri-facettes » selon (Kubicki 2006)*

Afin d'être médiatisé par l'outil, le contexte de l'activité doit être connu et caractérisé. Cela fait l'objet des études citées précédemment (cf. § 1.1).

Les rôles et expériences dites « métier » de l'acteur sont les facteurs déterminants de son contexte. Ils vont grandement influencer son besoin d'information et donc la manière dont il voudra et devra percevoir les éléments médiatisés. Par exemple, un ingénieur voudra avoir un accès aux documents plans afin de les consulter, les analyser, voire les modifier alors qu'un coordinateur cherchera avant tout à savoir s'ils ont bien été diffusés, dans les délais convenus, etc…

Enfin le contexte utilisateur c'est la combinaison des fonctionnalités techniques offertes par un outil et de l'environnement dans lequel se trouvent l'utilisateur et cet outil. Le projet AIC est un cas typique d'activité à environnements multiple : l'agence, le chantier, les partenaires, les lieux de réunions diverses… Les technologies à disposition sont de plus en plus performantes et accessibles et les possibilités d'innovation sont croissantes, comme le démontre l'expérience SDC que nous venons d'illustrer. Cela s'applique également et tout particulièrement aux technologies mobiles. Les performances des ordinateurs portables, des tablettes tactiles et Smartphones ont augmenté, tout comme leur autonomie et leur connectivité réduisant ainsi le nombre de contraintes que l'on pouvait assigner à la mobilité il y a quelques années (Satyanarayanan 1996).

## 1.3 Conclusion

Nous avons dans ce chapitre caractérisé la coordination d'un projet AIC et les outils qui la supportent. En conclusion de la première partie sur la caractérisation du contexte de l'activité collective (cf. 1.1.4), nous avons introduit le fait que celle-ci puisse servir la conception des outils qui le médiatise. Pour cela, nous avons choisi d'isoler l'analyse et la caractérisation de ces outils pour ensuite identifier ce qui définit la manière dont ils s'adaptent au contexte.

Des études ont montré qu'à ce contexte de l'activité collective se greffent deux autres contextes relatifs aux actions individuelles de l'acteur : le contexte acteur et le contexte utilisateur. Nous parlons alors d'un **contexte « tri-facettes »**.

C'est par rapport à l'ensemble du contexte tri-facettes que nous devons considérer la conception d'un outil car c'est cette combinaison qui définit le mieux les besoins des acteurs dans leur globalité. Or, la qualité perçue des outils est étroitement liée à leur capacité à répondre à ces besoins.

Dès lors, deux approches de conception se distinguent :

- une approche « top-down », selon laquelle on crée les outils en fonction des besoins (c'est un processus d'ingénierie classique),
- une approche « bottom-up » pour laquelle les outils sont créés dans un processus d'innovation technologique puis transférés vers un ou plusieurs secteurs d'utilisation avec une éventuelle adaptation (on parle de rétro-ingénierie).

Au cours du chapitre suivant, nous nous penchons sur l'analyse du développement d'une plateforme d'échange de documents particulière. L'objectif est de pouvoir en évaluer la méthode de conception afin de comprendre les points forts à capitaliser, mais aussi les limites à surmonter.

# Chapitre 2 – Un cas de développement d'un outil de TCAO : CRTI-weB

Le projet de recherche Build-IT fut mené par le Centre de Recherche Public Henri Tudor (CRP HT) et le Centre de Ressources des Technologies et de l'Innovation pour le bâtiment (CRTI-B) pour « guider le secteur luxembourgeois de la construction vers l'utilisation d'outils numériques d'assistance à la coopération » (Guerriero 2009). Basé sur l'analyse des « bonnes pratiques » du secteur, ce projet aboutit au prototypage d'un outil : CRTI-weB. Cet outil est composé de deux « services métier » : un service d'échange de documents et un service de gestion (édition et partage) de compte-rendu de chantier[6]. Ce que nous appelons « expérience CRTI-weB » comprend ce projet de recherche, mais aussi les développements, le transfert et les évaluations de l'outil qui ont suivi. Elle comprend également les travaux actuels pour l'amélioration de l'outil par la proposition de nouvelles fonctionnalités.

Le contexte de travail au sein du CRP Henri Tudor offre un statut d'observateur privilégié sur le déroulement actuel de cette expérience. Cela nous a permis d'analyser l'utilisation de l'outil ainsi que le travail de développement mené pour l'améliorer.

## 1.4 Description de l'expérience

L'approche menée a suivi une trame de développement basée sur la notion de « bonnes pratiques à outiller » que l'on peut découper ainsi : identification des besoins métiers et bonnes pratiques, caractérisation de l'information à manipuler, proposition et définition des services proposés par l'outil (pour chaque bonne pratique) et développement.

Devant la difficulté de déterminer précisément les fonctionnalités utiles à la collaboration considérant les facteurs à risques du projet (cités plus tôt en 1.2.1), la définition finale des bonnes pratiques à mettre en place a fait l'objet d'un consensus entre le CRP HT et le CRTI-B. En ce

[6] Plaquette descriptive de la plateforme disponible sur ce lien : http://uat.crti-web.com/public/Description_de_la_plateforme_CRTIweB.pdf

sens, ces pratiques consensuelles sont relatives à des comportements généralement observés dont les deux organisations ont jugé l'outillage nécessaire.

Nous introduisons ici le concept de service, jusque-là non évoqué. Cette notion que nous détaillerons plus tard permet de définir les fonctionnalités d'un outil, aussi bien d'un point de vue général que d'un point de vue technique.

### *1.4.1 Identification des besoins métiers et bonnes pratiques*

Les premières phases d'analyse sous la forme de brainstorming ont conduit à l'identification de deux enjeux ou besoins relatifs à l'outillage numérique de la conduite d'un projet AIC :

- la rédaction et la diffusion de comptes-rendus de chantier en prenant en compte les spécificités et la structure de ce document,
- la diffusion de documents diverses durant le projet, que ce soit des plans, des textes, des tableaux…

Relativement à ces besoins, les professionnels ont pu identifier certaines « bonnes pratiques » généralement adoptées dans le cadre de leur travail (Tableau 1). Une bonne pratique est caractérisée par une action (ex. rédiger, consulter…) qui « cible » une information (ex. le compte-rendu, les requêtes sur un plan…).

Tableau 1. Besoins métiers à outiller et bonnes pratiques liées

| Besoins | Bonnes Pratiques |
|---|---|
| Rédaction et diffusion de CR | (CR1) Rédiger un compte-rendu selon un modèle prédéfini |
| | (CR2) Animer la réunion de chantier et prise de notes |
| | (CR3) Consulter le compte-rendu |
| | (CR4) Lire les remarques qui nous concernent |
| | (CR5) Réactions sur les points particuliers du CR |
| | (CR6) Se tenir au courant de l'avancement des tâches de construction |
| | (CR7) Archiver un compte-rendu à valeur contractuelle (non-modifiable) |
| | (CR8) Rechercher les remarques en cas de litige |
| Diffusion de documents | (Doc1) Standardiser le nommage des documents |
| | (Doc2) Décrire et localiser les modifications effectuées sur un document |
| | (Doc3) Informer les personnes concernées d'un dépôt ou de la de la modification d'un document |
| | (Doc4) Transmettre et enregistrer les requêtes relatives à chaque document |
| | (Doc5) Réagir concernant un document et tracer les réactions |
| | (Doc6) Maîtriser la visibilité des documents |
| | (Doc7) Superviser et gérer l'échange des documents |

Ces bonnes pratiques sont exercées par un ou plusieurs acteurs en fonction de leur rôle à la fois organisationnel et opérationnel. Par exemple, la rédaction d'un compte-rendu est attribuée au coordinateur (rôle organisationnel) qui est alors considéré comme seul éditeur (rôle opérationnel)

de ce type de documents, alors que le partage de plans se fait par tous les concepteurs impliqués dans la production plans.

### *1.4.2 Caractérisation des comptes-rendus et autres documents*

La modélisation du contexte de l'activité collective dans un projet AIC a permis de définir un certain nombre de concepts, dont celui de documents. L'étude des besoins a montré que le compte-rendu faisait l'objet d'une caractérisation particulière, possédant une structure propre à considérer dans la définition d'un service dédié à sa gestion. Comme nous pouvons le lire dans (Guerriero 2009), un modèle spécifique a été défini « prenant en compte les concepts particuliers de gestion de chantier autour du compte-rendu, et de la gestion documentaire ».

Un compte-rendu est lié à une réunion de chantier, résumant les informations relatives à celle-ci et permettant de garder trace de ce qui s'y est dit. Une réunion de chantier est en général précédée d'une visite de celui-ci, permettant de relever les éléments dont il faudra discuter. Ainsi, le compte-rendu de chantier contient un ensemble d'informations structurées sous la forme de rubriques.

- Les références servent à l'identification d'un compte-rendu, comme un numéro, une date de création, le nom de l'auteur… Ces références sont notamment très utiles lors d'une recherche par filtrage de métadonnées.
- La liste de présence et de diffusion contient les personnes présentes à la réunion de chantier et à qui il faudra diffuser ce rapport.
- Les généralités font état de la situation du chantier et des ressources sur le site. Elles identifient notamment les interruptions dues aux conditions météorologiques.
- Les notes sont des règles à suivre et des clauses qui s'appliquent à chacun (elles sont définies en début de projet et ne changent en général pas en cours de projet).
- La liste des remarques soulève les points particuliers dont on aura parlé pendant la réunion. Les remarques possèdent leurs attributs propres, tels qu'un numéro, un intitulé, une description littérale, une priorité, une date de constat…« Ces remarques peuvent faire l'objet de rappels au fil des semaines, ou encore être illustrées par des croquis ou des photos de chantier ».
- L'avancement décrit la progression du chantier, il est notamment comparé au planning de chantier qui aura été fixé.
- L'agenda précise la date et l'heure dont on aura convenu pour la prochaine réunion.

Les autres documents sont les plans de tout type, les fiches techniques, les tableurs, les modèles, etc… c'est-à-dire tout document supportant l'échange d'information entre les acteurs du projet. Un certain nombre d'attributs les définissent et permettent leur gestion dans une base de données comme à travers l'outil CRTI-weB :

- un nom, qui lors de l'expérience crti-web a fait l'objet d'un travail particulier donnant lieu à une convention de nommage bien définie. Ainsi le nom de chaque document est la composition de plusieurs acronymes permettant d'identifier la phase de projet, le type de document, sa version, etc…
- un auteur, qui est identifié lors du partage du document après avoir été identifié,
- une zone de visibilité qui permet de restreindre l'accès à certains types d'acteurs,
- des actions et des réactions, qui permettent d'avertir un ou plusieurs acteurs de la disponibilité d'un document ou d'une tâche que celui-ci doit accomplir vis-à-vis de ce document.

Ces attributs servent à organiser l'information dans l'espace partagé et, au même titre que les références du compte-rendu, à les rechercher.

### *1.4.3 Services proposés*

Les fonctionnalités de l'outil proposé sont regroupées en deux services dits « services métier » que sont le service d'édition et diffusion de comptes-rendus et le service de partage de document, conformément aux deux besoins identifiés. Le concept de service métier est ici utilisé pour décomposer l'outil avec un haut niveau d'abstraction. Chaque bonne pratique est ensuite outillée par un « service informatique », une solution fonctionnelle pour effectuer cette pratique. Le tableau suivant décrit les services informatiques proposés pour chaque bonne pratique, dans les services métier « compte-rendu » (CR1 à 8) et « documents » (Doc1 à 7).

Tableau 2. Les 15 services informatiques composant l'outil CRTI-weB

| BP | Services informatiques |
|---|---|
| **(CR1)** | Rédaction |
| **(CR2)** | Génération d'un brouillon pdf |
| **(CR3)** | Consultation |
| **(CR4)** | Tri des remarques par organisme |
| **(CR5)** | Réactions |
| **(CR6)** | Notifications |
| **(CR7)** | Génération d'un Pdf sécurisé |
| **(CR8)** | Recherche |
| **(Doc1)** | Nommage de documents et convention de nommage |
| **(Doc2)** | Mise à jour |
| **(Doc3)** | Notifications |
| **(Doc4)** | Actions |
| **(Doc5)** | Réactions |
| **(Doc6)** | Zones de visibilité |
| **(Doc7)** | Tableau de bord des échanges |

L'ensemble de ces services informatiques fut implémenté sur base des technologies PHP/MySQL dans une architecture client/serveur et sont accessibles à travers une interface web. Ces services dits « services-web » sont décrits dans le protocole REST (Fielding 2000). Le chapitre 5 de ce manuscrit apporte plus de précisions sur le concept de service. Nous le réinterprétons également dans le cadre de notre proposition (chapitre 11).

Les deux figures suivantes illustrent l'interface des services métiers « documents » (Figure 11) et « comptes-rendus » (Figure 12).

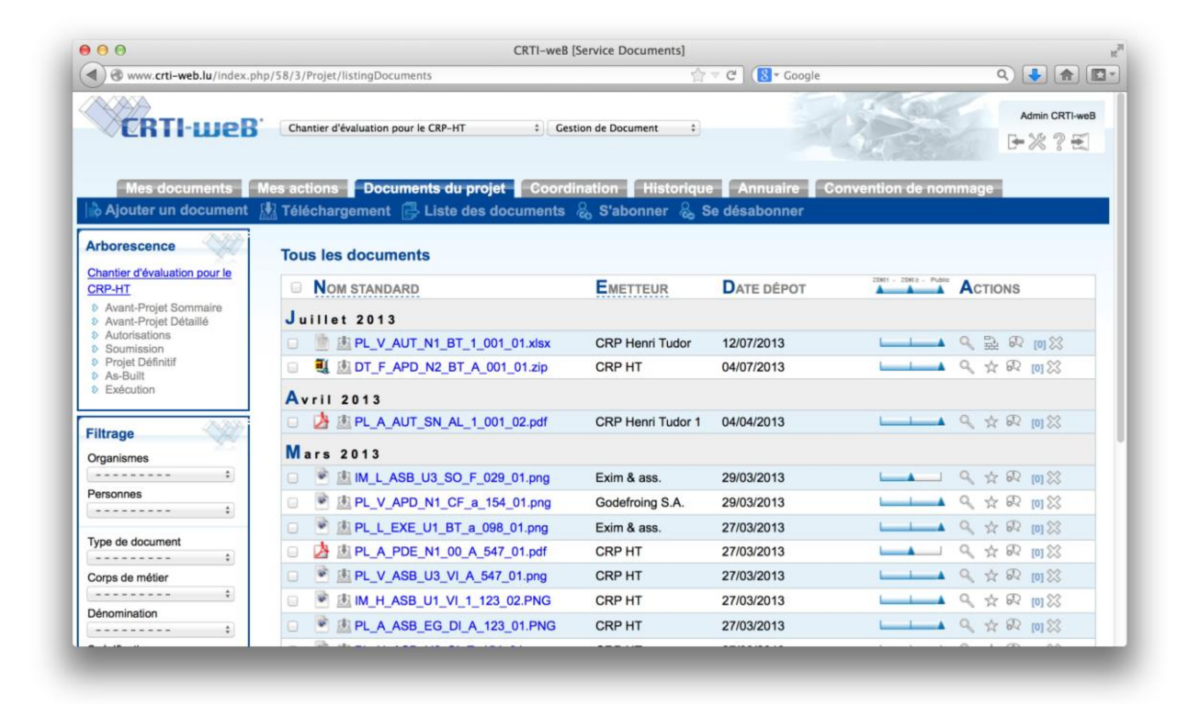



*Figure 11. Interface de CRTI-weB service « Documents »*

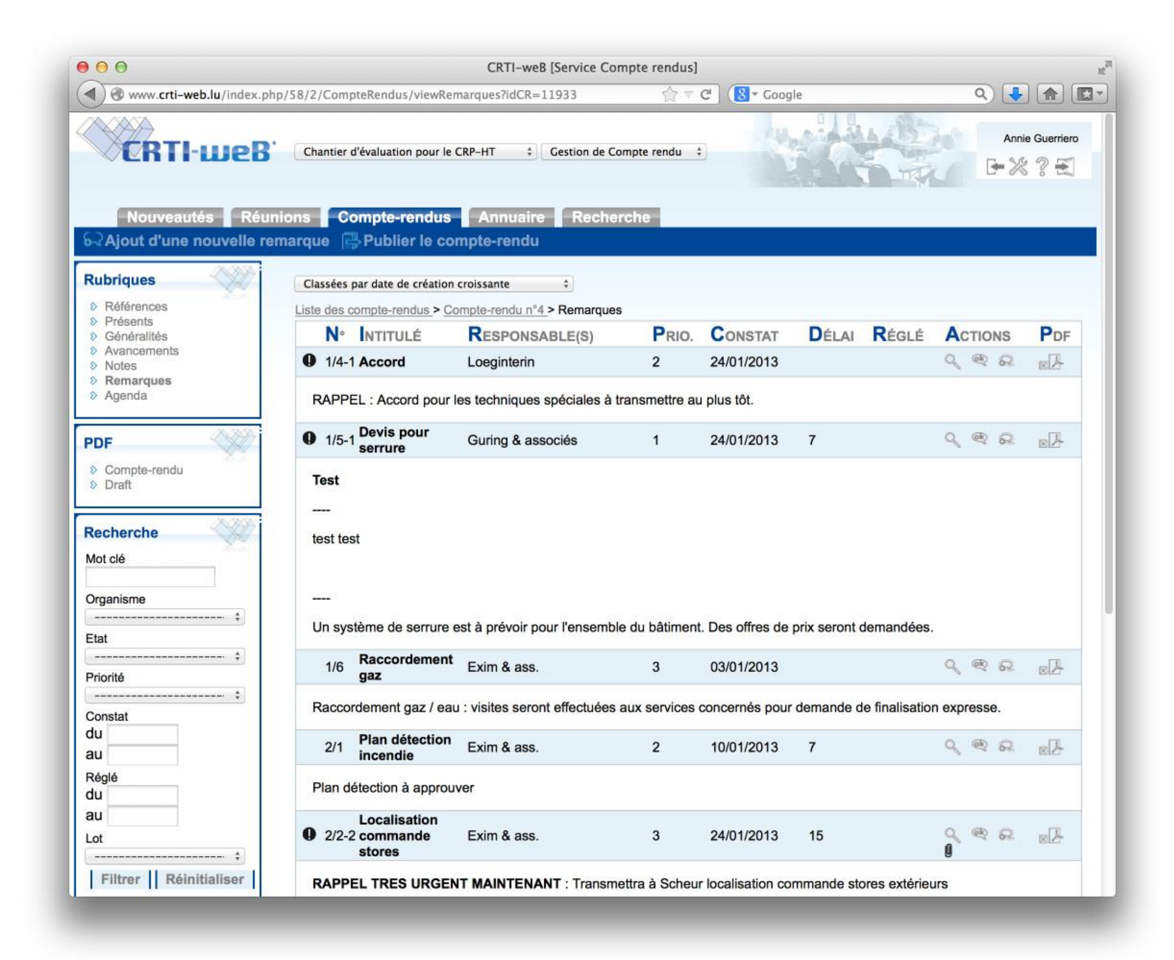



*Figure 12. Interface de CRT-weB service « Comptes-Rendus »*

## 1.5 Analyse critique

### *1.5.1 Contexte et protocole d'analyse*

Comme nous l'avons introduit plus tôt, ce travail de thèse s'est inscrit dans un contexte d'analyse et de développement autour de l'outil CRTI-weB.

En ce qui concerne l'analyse des utilisations, les retours d'expérience rassemblés par mes prédécesseurs ont composé un premier corpus d'éléments d'analyse. Nous les avons complétés en accédant en tant qu'observateurs aux différents projets sur la plateforme afin d'analyser précisément les partages. Nous avons notamment porté notre attention sur : l'objectif de ces partages, les auteurs en fonction des types de documents, l'utilisation des actions et l'utilisation des réactions ainsi que leur contenu.

Pour ce qui est de l'analyse du développement, nous avons pu porter un regard sur l'ensemble des tâches de développement prévues et menées (sous forme de « tickets » gérés par un outil en ligne) et identifier la nature de ceux-ci. En plus de cela, nous avons été impliqués dans une tâche de développement précise en tant qu'expert métier. Ce rôle d'expert a pour but d'analyser et comprendre les besoins des professionnels d'un domaine afin de les capitaliser et les transférer pour la conception de solutions adaptées.

### *1.5.2 Analyse de l'utilisation*

L'outil CRTI-weB a fait l'objet de plusieurs tests, du cadre d'évaluation de prototype à celui de feedbacks utilisateurs en utilisation « réelle » dans des projets AIC ou lors d'ateliers de formation. Le public évaluateur est composé à la fois de professionnels du secteur (architectes, entreprises de construction et Maîtres d'ouvrage) et des étudiants en architecture des universités de Nancy et Liège (les étudiants ont évalué uniquement le service « documents »). Les premiers résultats sont publiés dans (Guerriero 2009; Kubicki et al. 2009) mais d'autres enquêtes ont donné lieu à de plus récents constats sachant que l'outil fait l'objet d'améliorations constantes.

L'évaluation du service « Comptes-rendus » en situation d'atelier montre que l'outil apparaît fiable, facile à utiliser, compatible avec la pratique professionnelle et que l'apprentissage ne nécessite pas d'investissement personnel important. Pourtant en situation réelle il ne trouve pas sa place. Une des réponses formulées est largement liée à la difficulté de modifier les usages internes. En effet, il semble que les personnes aient adopté l'utilisation systématique des modèles (« templates ») Word ou Excel et que, bien qu'elles soient conscientes du potentiel de l'outil CRTI-weB « Compte-rendu », elles se satisfont de leur outil actuel.

Le service « Documents » a fait l'objet d'une analyse plus détaillée avec un retour sur chacun des services informatiques qui le composent. Des expérimentations ont permis d'observer les différences d'appréhension d'un public à un autre, à savoir entre professionnels et étudiants. Le premier contexte est basé sur des relations contractuelles et une rigueur appliquée à un grand nombre d'acteurs et de documents, tandis que le deuxième est informel et basé sur l'ajustement mutuel d'un petit groupe produisant peu d'information.

- De manière générale, le partage de documents doit être amélioré afin de prendre en compte la pluralité des formats d'un document (ex. un plan aux format dwg et pdf) et les liens qui

peuvent exister entre plusieurs documents. Le partage de documents « un par un » trouve vite ses limites dans des situations comme les rendus de projet dans lesquelles on partage beaucoup d'information (que ce soit dans le domaine professionnel ou étudiant).

- Le service de nommage est pertinent par rapport aux habitudes générales des professionnels. Il doit être cependant flexible, car la convention de l'Ordre de Architectes et Ingénieurs (OAI) peut ne pas être adoptée par tout le monde. Les étudiants sont, à l'inverse, réfractaires à ce système : vis-à-vis du temps supplémentaire passé à nommer chaque document, ils ne perçoivent pas de plus value, car le petit nombre de documents partagés ne requiert pas une telle structuration. Il en va de même pour le service « zones ».
- Les notifications par mail permettent d'éviter la consultation systématique de la plateforme pour surveiller les partages, mais peuvent vite devenir problématiques en cas de nombreux partages (surcharge de la boîte mail), particulièrement donc en milieu professionnel. Cela soulève la question de les regrouper dans des mails « résumé » à intervalles réguliers.
- Les réactions sur documents sont parfois détournées pour simplement stipuler que le document a été consulté ou pour faire référence à un autre document (cf. besoin de lier les documents entre eux). Les validations sont parfois faites informellement (en réunion, au téléphone) et non tracées sur la plateforme. Chez les étudiants, le caractère non hiérarchique du projet ne pousse pas à la demande de validations au profit de la recherche de conseils, et d'avis partagés. Se pose aussi la question de la visibilité des réactions vis-à-vis de tous les utilisateurs.

Ces éléments peuvent être interprétés comme des « contre-exemples », témoignant des lacunes de l'outil dans l'adaptation de l'outil.

### *1.5.3 Analyse du développement et conclusion*

Les études sur CRTI-weB nous permettent de relever plusieurs éléments importants relatifs quant au développement d'outils logiciels et plus particulièrement d'assistance au travail collaboratif/coopératif.

<u>Les exigences</u>

Comme nous avons pu le vérifier en analysant les tâches de développement initiées sur la plateforme après son transfert (120 tâches analysées, voir en annexes Figure 143, Figure 144, Figure 145, Figure 146), la moitié d'entre elles ont pour point de départ un « besoin métier », comme « partager des comptes-rendus de chantier avec photo », « connaitre l'activité d'un collaborateur » ou « récupérer toutes les données en fin de projet ». L'autre moitié des développements relève de l'implémentation de fonctionnalités ou de l'amélioration des fonctionnalités existantes pour une utilisation plus rapide ou « confortable ».

Considérant aujourd'hui l'adoption grandissante de CRTI-weB dans les projets de construction luxembourgeois, la définition et l'outillage des « bonnes pratiques » étaient un premier pas cohérent et nécessaire. Mais nous comprenons que malgré la rigueur du travail de conception basé sur l'étude de ces bonnes pratiques, et du fait du consensus autour des spécifications fonctionnelles à mettre en place, tous les besoins n'ont pu être pris en compte et devront l'être au fur et à mesure pour assurer l'adoption et la pérennisation de l'outil. Nous constatons également que au-delà de répondre à des besoins métiers, l'outil doit être agréable et simple à utiliser. (Guerriero 2009) cite (Davis 1989) en disant que « l'appropriation d'une nouvelle technologie repose sur deux dimensions » : l'utilité perçue et la facilité d'usage. Notre analyse confirme l'adéquation entre ces deux dimensions.

L'étude des usages de la plateforme nous montre que l'utilité perçue n'est pas absolue. Elle fait ressortir la relativité de cette utilité par rapport aux contextes identifiés plus tôt, dans la mesure où :

- les utilisateurs ont reconnu pour la plupart la pertinence des services proposés, montrant que l'analyse du contexte de l'activité a permis de répondre à des besoins existants,
- la non-adoption dans certains cas des services est due à un contexte utilisateur déjà trop fourni en outils et solutions largement adoptées,
- les variations entre professionnels et étudiants montrent l'influence du contexte de l'acteur sur l'appréhension d'un outil.

Au regard de cette analyse, nous nous questionnons alors sur la manière dont on pourrait à présent :

- améliorer l'analyse des besoins afin de prédire les futures demandes de développement,
- et prendre en compte les contextes utilisateur et acteur pour spécifier des services réellement adaptés à (et adoptés par) chacun.

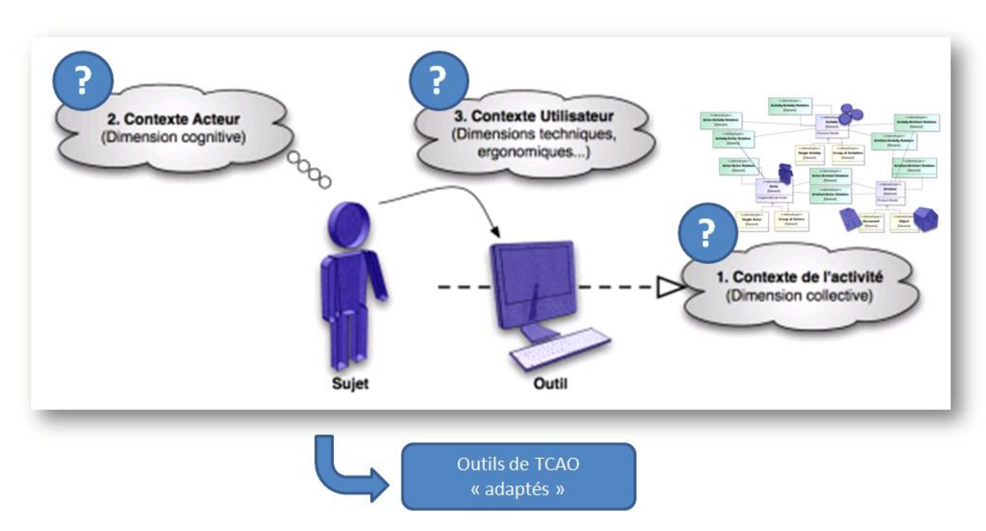



*Figure 13. Vers des outils de TCAO adaptés au contexte « tri-facettes »*

<u>La méthode de développement</u>

Les premières versions de l'outil ont été développées « en interne » au CRP Henri Tudor qui a également assuré un rôle d'expert métier en étroite collaboration avec les praticiens du domaine. Dès le transfert vers le secteur professionnel, les développements furent confiés à une société de services externe (Kitry Consulting[7]). Dès lors, le processus est passé dans un schéma collaboratif avec une réelle répartition des tâches de conception, développement, et d'évaluation entre acteurs différents.

Dans cette activité de développement, qui n'est pas sans rappeler l'activité collective d'un projet AIC, la coordination est également un enjeu. Depuis 2009, la société externe commercialise la solution et se charge également de la formation et du support des utilisateurs. Lorsque des modifications purement fonctionnelles doivent être réalisées, ils développent directement une solution, mais lorsque cela relève d'un problème métier, un processus de conception est de nouveau mis en œuvre, basé sur l'analyse des besoins et supporté par les experts métier. C'est

[7] http://www.kitrygroup.com/

d'ailleurs ce rôle que nous avons tenu au cours de notre implication dans le projet, ce qui nous a fourni un contexte d'observation privilégié.




*Figure 14. Schéma du processus de conception développement et transfert des services de CRTI-weB*

Nous nous questionnons dès lors sur la dépendance entre la qualité des solutions développées et la qualité de la coordination au sein de ce processus de conception. De manière générale :

- la qualité d'une solution en termes de services informatiques sera évaluée par les utilisateurs qui jugeront de la réponse de la capacité de ce service à répondre à leur besoin,
- la qualité de la coordination sera déterminée par la capacité des différents acteurs du projet à exprimer leur point de vue (Hanrot 2005) et à répondre aux attentes de chacun.

### 1.6 Conclusion

Les observations que nous venons de faire et les remarques soulevées nous permettent à présent de cadrer les objectifs relatifs à l'amélioration de CRTI-weB et, de manière générale au développement d'un outil de TCAO.

Ce type de développement doit être guidé par des exigences clairement identifiées et justifiées. En effet, il ne s'agit pas de développer des services que l'on devra modifier voir supprimer dans un second temps, car ils seront inutilisés. Ce genre de méthode s'avère trop longue et couteuse. La gestion du projet de développement est également un enjeu important. Il s'agit de bien transmettre l'information nécessaire à la réponse du besoin et d'adopter une approche méthodique pour le développement de la solution.

Le chapitre suivant approfondit ces deux enjeux pris comme point de départ pour la définition de notre problématique de recherche ainsi que l'identification des axes d'étude à analyser.

# Chapitre 3 - Problématique et méthode de travail

Nous définissons dans ce chapitre la problématique qui émerge des contextes d'étude et objectifs relatifs identifiés précédemment : la proposition d'une méthode de conception de services adaptés au secteur de la construction. La méthode de travail suivie pour répondre à cette problématique est présentée sur la base d'un recul théorique sur la science de la conception.

## 1.7 Construction de la problématique

### *1.7.1 Objectifs et concepts vis-à-vis du contexte tri-facettes*

Notre problématique se construit dans un premier temps autour de l'analyse et la critique du contexte tri-facettes introduit par (Kubicki 2006). L'objectif de ce travail était la multi-visualisation du contexte de l'activité, dans le but de favoriser les conditions d'une coordination basée sur l'ajustement mutuel.

Suivant nos premières remarques, ce contexte sert de base à la caractérisation des besoins métiers des acteurs. Les artefacts, regroupant documents du projet et ouvrages y sont bien définis, mais nous constatons que la définition des activités reste générique. Les relations acteurs-artefacts et acteurs-activités caractérisent les actions de chacun que ce soit d'un point opérationnel (acteur-artefact) ou organisationnel (acteur-activité), cependant ces concepts ne sont pas mis en avant. Il n'y a d'ailleurs pas de liste exhaustive des types de ces relations ou d'attributs déterminants les caractérisant. Nous avons donc cherché à améliorer notre perception du contexte de l'activité en améliorant la définition de ces relations.

Vis-à-vis des analyses sur la coordination, nous partons du postulat que l'activité de projet doit aussi bien être définie d'un point de vue collectif qu'individuel. Pour cela il nous semble important de prendre également en compte le profil métier et les préférences d'un acteur, ou en d'autres termes le « contexte acteur » qui est déterminant dans la réalisation d'une activité individuelle.

Le troisième contexte à prendre en compte est le « contexte utilisateur ». Nous devons donc aborder un point de vue technique. Ici nous remettons directement en cause la place de l'outil dans le contexte de l'activité collective. En effet, nous supposons que l'outil « supporte » le contexte de l'activité collective en le médiatisant : il aide les acteurs à réaliser leurs activités et à gérer les artefacts du projet. Il est vu comme un « moyen » et non comme un devoir. Prendre en compte le contexte utilisateur consistera donc à comprendre les possibilités techniques dues à l'outil et son environnement, et à percevoir en quoi il supporte une activité, et dans quelle mesure il est perfectible (ou remplaçable).

Dès lors, plusieurs concepts apparaissent comme importants. En voici une brève introduction, nous les approfondissons dans les différents chapitres de l'état de l'art (chapitres 4 à 7) pour ensuite les intégrer dans notre approche (chapitres 8 à 11).

Le concept de pratique

Les activités de projet sont des activités créatrices. En d'autres termes, elles ne relèvent pas de la théorie mais de la pratique. Nous venons d'introduire, lors de la description de l'expérience CRTI-weB, le concept de « bonnes pratiques ». Elles font référence ici à des « astuces » (Chen 2009), à des pratiques « qui ont fait leurs preuves » comme par exemple la réunion hebdomadaire de chantier suivie de la rédaction d'un rapport de chantier. Selon (Clot 2007) la bonne pratique serait « la transformation collective – institutionnellement secondée – de l'activité en instrument d'une autre activité ». Dans CRTI-weB, ces bonnes pratiques devaient en plus être consensuelles.

Nous retenons ce point de vue par lequel **l'activité de projet est « instrumentée » par d'autres activités que sont les pratiques**. Cependant, le concept de « bonne pratique » n'est pas universel, car il peut être dépendant du contexte du projet (c.à.d. on admet qu'une pratique dite bonne dans un projet pourra ne pas être adaptée dans un autre). Ce contexte est relatif à la nature du projet et à un « ensemble de contextes » que sont le contexte géographique, culturel, économique, social, technologique…, autant de facteurs qui influenceront la manière d'appréhender le projet et donc ses pratiques.

Cela nous amène à opérer une distinction entre les pratiques génériques (que l'on retrouve dans tous les projets AIC) et les pratiques spécifiques (qui sont propre à un contexte de projet particulier). Nous distinguons également les pratiques collectives (effectuées à plusieurs) des pratiques individuelles. Les chapitres 5 et 9 illustrent notre acception et le positionnement de notre approche par rapport au concept de pratique.

Les processus de l'information et du matériel

(Björk 1999) divisent le processus de construction d'un bâtiment en deux sous-processus relatifs au cheminement de l'information et du matériel dans un projet. Un sous-processus (aussi appelé activité) est caractérisé par des Inputs, des Outputs, des Ressources. On retrouve en tant qu'inputs et outputs les artefacts définis précédemment, à savoir les documents et les objets. Les ressources sont les acteurs responsables de ces activités, les outils ainsi que la connaissance utilisés.

En ce qui concerne les sous-processus d'information, Björk considère ces activités dites « primaires » comme des règles supportées par des activités secondaires dont il en distingue quatre types :

- la production d'une nouvelle information,
- la communication de personne à personne,
- la diffusion d'information,
- la recherche et la récupération d'information.

Le chapitre 10 présente comment nous intégrons ces concepts afin d'approfondir notre description du métier.

Le concept d'usage

Nous préfèrerons utiliser le concept d'usage plutôt que celui d'utilisation lorsque nous voudrons considérer le rapport d'un acteur vis-à-vis de l'outil. Cette distinction va au-delà d'un choix sémantique et trouve notamment sa justification dans le domaine du génie logiciel (Constantine & Lockwood 2003). Le chapitre 4 introduit les différentes approches de conception logicielles dont la conception centrée-usages. Le chapitre 10 (et particulièrement le § 1.31.1) approfondit la distinction entre usage et utilisation et positionne notre approche par rapport à ce concept.

L'objectif au travers du concept d'usage et de décrire l'acteur en tant qu'utilisateur, le contexte d'utilisation dérivé du contexte métier et finalement l'outil lui-même comme médiateur de l'activité.

Les services

Comme l'illustre l'expérience CRTI-weB, un outil peut être décomposé en plusieurs niveaux de services. La qualité de ces services au regard des professionnels d'un secteur conditionne l'adoption d'un outil par ces derniers.

Le modèle de qualité de services proposé dans (Bjekovic & Kubicki 2011) repose sur six critères :

- la correspondance au métier qui implique la réponse à une problématique métier, avec une amélioration des pratiques et par extension la bonne réputation du service dans le milieu,
- la stabilité qui est relative à la justesse du service (la pertinence de l'information produite), sa précision et sa disponibilité,
- la performance en termes de temps de réponse et d'utilisation des ressources,
- la sécurité assurée par le respect de la confidentialité, l'intégrité (c.à.d. la prévention des accès et modifications), la traçabilité et l'authenticité,
- l'utilisabilité, synonyme de compréhension, d'intuitivité, de facilité d'utilisation et de prévention des erreurs vis-à-vis de l'utilisateur,
- le respect de l'existant, de par le support des standards et l'interopérabilité avec les services existants.

Afin d'atteindre ces critères, il est nécessaire de structurer le processus de conception des services. Il s'agit typiquement de fournir une information pertinente aux équipes et d'assurer

l'efficacité et la traçabilité des échanges. Nous introduisons les services collaboratifs dans le chapitre 6 et les intégrons dans notre approche dans le chapitre 11.

### *1.7.2 Formulation de la problématique*

À partir des éléments introduits dans les chapitres précédents, nous sommes en mesure de déterminer les axes de recherche à investiguer dans la suite de ce travail : l'analyse des besoins et la conception logicielle.

La question que nous nous posons est la suivante :

Pouvons-nous proposer une méthode structurée et basée sur l'analyse du contexte tri-facettes pour le développement (ou la mise à jour) de services informatiques supportant les besoins de collaboration (ou « services collaboratifs ») propres au projet AIC ?

Avant de nous pencher vers l'état de l'art sur les méthodes et les cas pratiques de conception logicielle et plus particulièrement de services collaboratifs, nous présentons dans le sous-chapitre suivant notre méthode de recherche introduite par une approche théorique sur la conception. Nous souhaitons en effet nous inscrire dans une démarche scientifique qui nous permette de généraliser notre approche, notamment vis-à-vis des éventuels besoins d'autres domaines que le projet AIC.

## 1.8 Méthode de recherche

Au regard de la problématique formulée, nous avons mené une étude sur les processus de conception. Notre recherche s'est basée sur des allers-retours entre théorie et cas concrets qui sont justifiés par les contextes d'études présentés au cours des deux premiers chapitres. En effet :
- ce travail possède un caractère appliqué, grâce au contexte de développement de CRTI-weB et son transfert,
- il appartient à un contexte scientifique particulier, entre le monde académique (à savoir les thématiques du CRAI) et les projets d'innovation (enjeux du CRP Henri Tudor).

À partir d'une étude sur les différentes approches de modélisation d'un processus de conception, (Laaroussi 2007) décrit un processus générique qui « en tant que processus cognitif de résolution de problèmes, s'articule autour de trois activités primitives : l'analyse du problème, la proposition d'une solution et l'évaluation de cette solution ». Nous nous sommes tournés vers les théories de la science de la conception pour comprendre la nature de ce processus.

### *1.8.1 La science de la conception*

On définit la science de la conception comme « science de l'artificiel » (Simon 2004) ou « science de la pratique » (Gregor 2009). Elle traite de la création d'objets et de savoirs par l'Homme pour atteindre certains buts (ex. l'ingénierie, l'architecture…), contrairement aux « sciences comportementales » (Hevner et al. 2004) à savoir les sciences naturelles (ex. la physique, la chimie…) et humaines (ex. l'histoire, la linguistique…) qui visent à expliquer des éléments existants de la nature ou de la société.

La science de la conception consiste en la construction et l'évaluation de produits de quatre types : des concepts, des modèles, des méthodes et des instanciations (March 1995) :

- les concepts « forment le vocabulaire d'un domaine »,
- un modèle décrit une situation en exprimant les relations entre les concepts,
- une méthode est une suite d'étapes suivies pour accomplir une tâche,
- une instanciation est la réalisation d'un artefact dans son environnement.

Atteindre un but, un objectif ne se fait pas sans contraintes, qu'elles soient physiques (contraintes techniques, géographiques, météorologiques…) ou morales (en termes de qualité, de délai, de prix…). Concevoir c'est donc obtenir une solution qui satisfasse un maximum de ces contraintes, ce qui implique de les connaitre, de les évaluer voire de les prioriser.

Comme on peut le voir dans (Peffers et al. 2007), les modélisations de l'activité de conception sont variées, mais suivent une trame commune. Nous retenons l'approche de (Vaishnavi & Kuechler 2007) qui nous apprend que la conception est non seulement basée sur un processus, mais aussi sur la production d'information et d'artefacts et la génération de flux de connaissances (Figure 15). Le processus défini par (Vaishnavi & Kuechler 2007) contient cinq phases.
- Connaissance du problème : cette phase fait ressortir un besoin et fixe les objectifs et les contraintes.
- Suggestion : c'est la phase créative qui suit directement l'expression de l'idée. En ressortent « un premier essai », une « esquisse » de solution.
- Développement : la suggestion est approfondie et implémentée. Les techniques utilisées et les artefacts créés sont dépendants du domaine.
- Évaluation : l'artefact créé à la phase précédente est évalué en fonction des objectifs fixés en première phase.
- Conclusion : il s'agit de dresser le bilan et de capitaliser les résultats

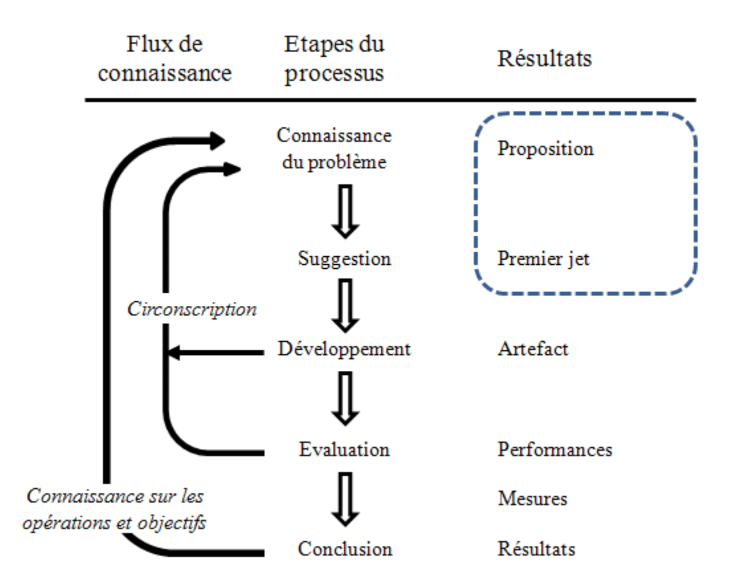



*Figure 15. Processus de conception, flux de connaissances et résultats (tiré de (Vaishnavi & Kuechler 2007))*

Nous remarquons que chaque étape d'un processus de conception génère un flux de connaissance qui alimente la connaissance globale du problème, encourageant le raffinement des choix au fur et à mesure des itérations de ce processus.

### *1.8.2 « Concevoir une méthode de conception »*

À partir des éléments qui composent notre problématique (voir 1.7.2) nous constatons que celle-ci relève de la science de la conception sous tous ses aspects :

- nous voulons concevoir des services informatiques pour le Travail Collaboratif Assisté par Ordinateur (le produit souhaité est une **instanciation**).
- Nous voulons que ces services soient adaptés à un domaine, ce qui nécessite d'en construire les **concepts**.
- Nous voulons concevoir une **méthode** qui supporte les objectifs précédents.
- Nous voulons construire des **modèles** sur lesquels s'appuie la méthode.

La figure suivante (Figure 16) illustre notre approche et la structure des prochains chapitres de cet ouvrage. La partie 2 (chapitres 4, 5, 6 et 7) propose une analyse des méthodes de conception dans divers domaines du développement logiciel et ce que nous en retenons pour « concevoir notre propre méthode de conception ». La description de cette méthode et son application dans divers cas d'étude font l'objet de la partie 3 (chapitres 8, 9, 10, 11 et 12).

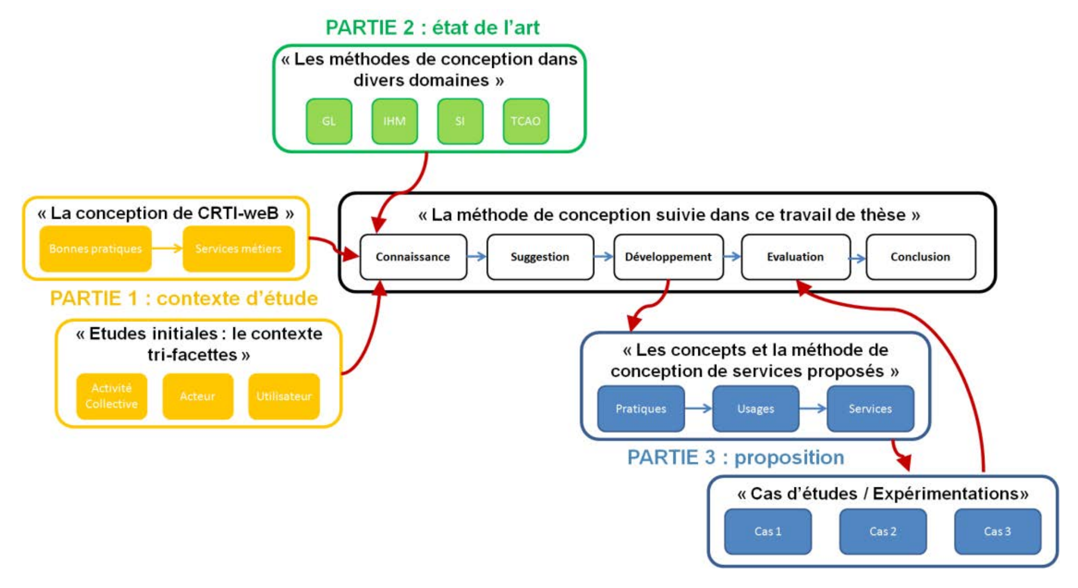



*Figure 16. Démarche suivie dans notre recherche basée sur la conception*

# PARTIE 2 : Théories et méthodes. Concevoir des services collaboratifs adaptés

La précédente partie fait émerger la nécessité de concevoir et proposer des solutions informatiques adaptées aux pratiques d'un métier en nous intéressant plus particulièrement aux pratiques de collaboration dans un projet AIC. L'enjeu de cette étude est de définir une méthode qui puisse guider cette conception. La partie 2 de ce manuscrit présente les théories et méthodes que nous avons explorées pour constituer notre base de connaissance sur le sujet. Nous avons notamment porté notre attention sur l'importance de la modélisation les différentes approches relatives. Nous nous sommes tout d'abord intéressés aux méthodes de conception logicielle et d'IHM (chapitre 4). Nous avons ensuite exploré le concept de services et la conception des services informatiques dits « de technologies de l'information et de la communication» (ICT) dans l'entreprise (chapitre 5). Nous avons enfin étudié des principes de conception de collecticiels et services collaboratifs (chapitre 6). Le dernier chapitre de cette partie (chapitre 8) illustre l'analyse et la critique de trois cas d'étude de conception particuliers. Il se conclut par l'évocation des éléments majeurs retenus comme base de notre propre méthode.

# Chapitre 4 - De l'utilisateur à la conception logicielle et d'interfaces

Ce premier chapitre retrace l'évolution de la conception logicielle et en décrit les activités relatives. Il a notamment pour but de souligner l'importance de l'analyse des besoins des utilisateurs pour la spécification de logiciels et d'interfaces homme-machine (IHM) et d'illustrer la place des modèles dans une telle approche.

## 1.9 Les méthodes et activités relatives à la conception logicielle

La conception logicielle s'inscrit dans une approche classique de conception telle que décrite précédemment. Chaque phase apporte un élément important dans le processus de la conception et sera déterminante dans la qualité du résultat et l'atteinte des objectifs. Comme le dit (Sommerville 1996) cité dans (Dibbern et al. 2009) (p.13), « l'enjeu de l'ingénierie logicielle est de permettre la conception de logiciels de qualité par **l'utilisation de techniques (ou processus) et d'outils appropriés** ». Dans le cadre d'une organisation, la conception logicielle consiste aussi à comprendre comment les systèmes « homme-machine » produisent et diffusent de l'information et influencent les organisations dans lesquelles ils sont intégrés (Vaishnavi & Kuechler 2007). Le cycle de vie du développement d'un logiciel comprend non seulement sa conception, mais aussi son évaluation, sa distribution et sa maintenance.

Tel que (Booch et al. 2007) le définit, **une méthode de conception logicielle** c'est un **processus** qui vise à produire une suite de **modèles**[8] décrivant, avec une **notation (ou formalisme)** propre, différents aspects d'un logiciel en cours de développement. Il ne s'agit pas dans cette étude de décrire toutes les méthodes de conception logicielle, mais plutôt d'en comprendre les concepts importants. En d'autres termes, nous cherchons davantage ici à comprendre la méthodologie de la conception logicielle afin d'y positionner notre propre méthode, en nous demandant notamment :

[8] Un modèle est la représentation de quelque chose, sous forme d'objet physique (ex. une maquette) ou de description (un texte, un dessin…). Un modèle peut servir à la description (ex. le plan d'une ville) comme à la conception (ex. le plan d'une maison à construire).

- comment se déroulent les processus ?
- Quels sont les modèles créés et utilisés ?

### *1.9.1 L'évolution des processus de conception*

Les méthodes traditionnelles de conception logicielle ont connu leur essor dans les années 60-70, période de démocratisation (grâce à la baisse des prix) mais aussi de croissance en termes de performances des outils informatiques. Les premières étaient des méthodes « en cascade » et suivaient un processus linéaire (Figure 17, partie gauche). Puis elles adoptèrent une structure en 2 étapes (dite en V) : le processus de développement dans une approche « top-down », se déroulant de l'analyse des besoins vers le code implémenté et le processus de test qui « remontait » du code vers les besoins (Figure 17, partie droite). On les a aussi qualifiées de méthodes orientées composants, car elles introduisaient le développement par assemblage de composants logiciels « préfabriqués » s'inspirant des domaines de l'électronique ou de la mécanique (Crnkovic et al. 2005; Bose 2002).

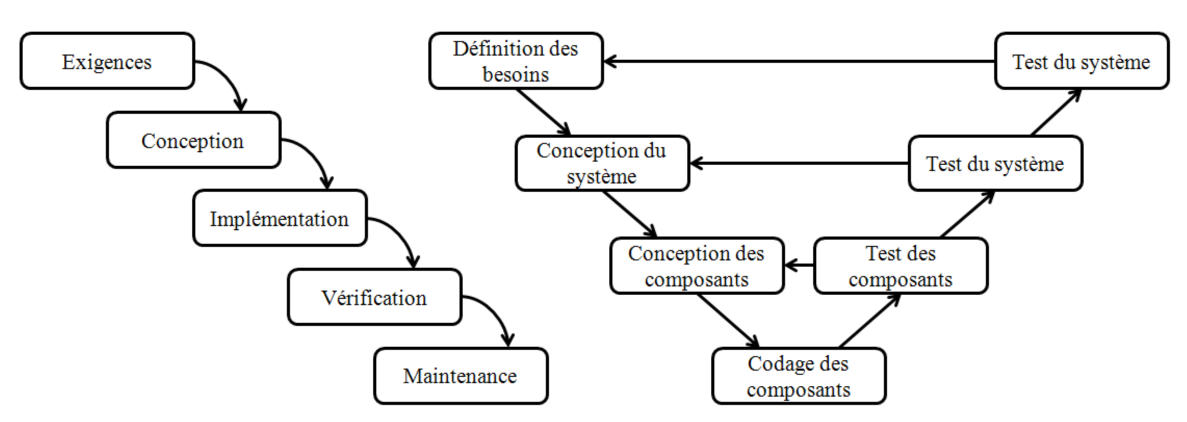



*Figure 17. Structure des méthodes traditionnelles de conception logicielle*

Le modèle en spirale (Boehm 1988) introduit une variante dans les méthodes de développement précédemment présentées (Figure 18). En effet après une phase de détermination des objectifs, contraintes et alternatives (quart supérieur gauche), on identifie et cherche à résoudre les risques relatifs aux objectifs. Cela se traduit par un premier prototypage et une évaluation de celui-ci (quart supérieur droit). Si ce premier prototypage ne permet pas de lever les risques, alors on planifie une seconde phase (quart inférieur gauche) et réitère le prototypage jusqu'à obtenir un prototype opérationnel (déplacement vers la droite dans le quart supérieur gauche) : on parle alors de conception dirigée par les risques. Cependant si une des phases précédentes de prototypages permet de lever les risques majeurs, les prochaines étapes suivent le cheminement classique du développement en cascade (quart inférieur gauche).

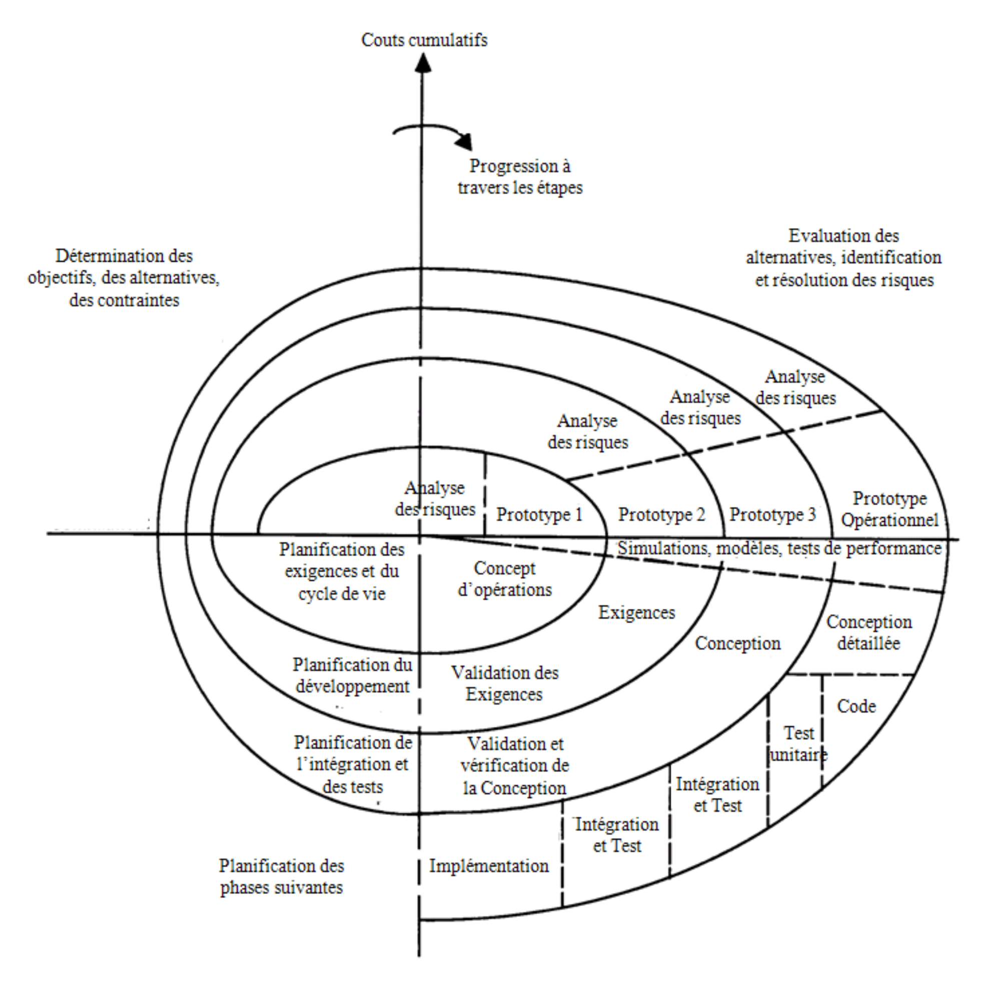



*Figure 18. Schéma de la méthode en spirale*

### *1.9.2 Vers un processus itératif*

Lors d'une validation ayant lieu en fin de projet, toute évaluation négative aura un coût conséquent, car impliquant un « retour à la case départ » dans le processus de conception. C'est pourquoi les méthodes plus récentes de développement logiciel dites « livré clé en main » tel que le « Processus Unifié de Développement Logiciel » (USDP pour Unified Software Development Process) puis le RUP pour (Rational Unified Process) se sont basées sur un processus déployant plusieurs cycles de développement dans le temps du projet de conception : un processus itératif et incrémental (dont le nombre d'itérations pourra varier selon le type et l'ampleur du projet). Chaque cycle de ce processus comprend quatre phases de développement (Kruchten 2001; Kroll & Kruchten 2003) et se termine par un jalon, à savoir sa propre évaluation. Les tests qui supportent ces évaluations peuvent être déterminés en début de cycle comme le préconise le processus de développement « dirigé par les tests » (K Beck 2003).

- L'étude préliminaire (ou « inception ») a pour but d'avoir une vue d'ensemble des objectifs, des couts, des risques.
- L'élaboration vise à décrire plus précisément les besoins, décrire le système, lever les risques et planifier le projet.
- La construction n'est autre que la proposition de la solution, sous forme fonctionnelle ou sous forme de documentation, selon le moment dans le projet.
- La transition vise à déployer et tester la proposition voire à la livrer si le cycle se situe en fin de projet, auquel cas cela comprend aussi la formation des utilisateurs et la maintenance.

Chaque phase fait appel à une ou plusieurs activités dont les principales sont les suivantes (voir Figure 19) : la modélisation du métier, la définition et la gestion des exigences, l'analyse et la conception, l'implémentation, les tests et le déploiement (Kruchten 2001). Ces activités sont relatives à des **disciplines** différentes qui impliquent des acteurs différents.

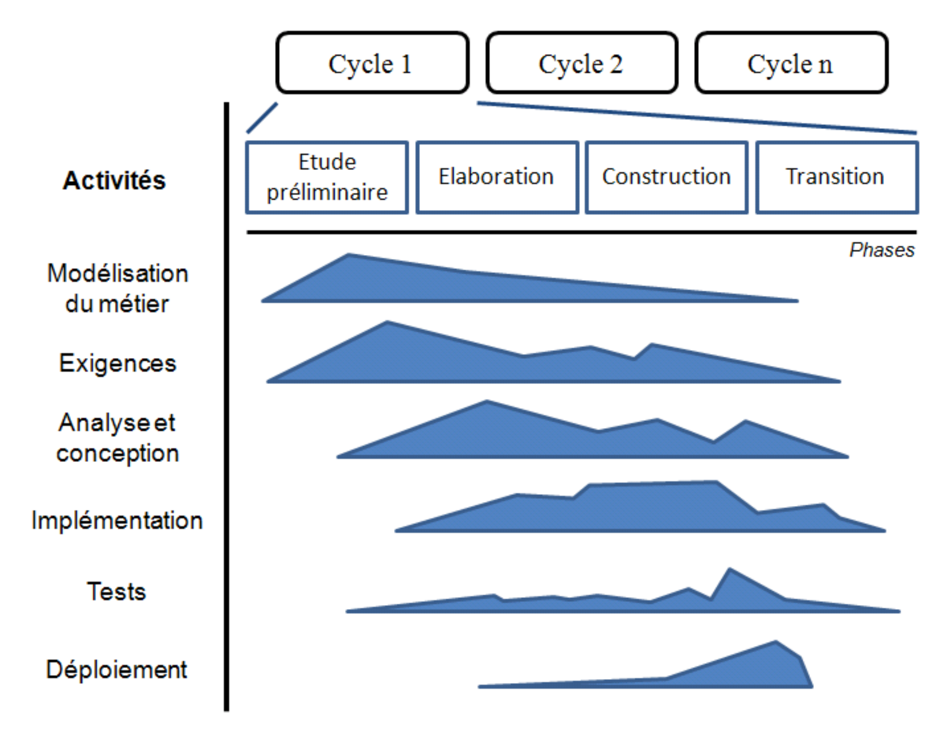



*Figure 19. Processus, cycles, phases et activités d'un projet de développement logiciel*

Ces activités s'appuient en partie sur la réalisation et l'utilisation de différents modèles dont les modèles UML (pour Unified Modeling Language). Ces modèles décrivent le système selon différents points de vue.

### *1.9.3 Les activités de développement et leurs modèles*

La modélisation métier

Comme on peut le lire dans (Gabay & Gabay 2008), la modélisation du domaine (ou modélisation métier) « permet de mieux comprendre la structure et la dynamique de l'organisation étudiée. Elle assure au client que les utilisateurs finaux et les développeurs partagent une vision commune de l'organisation ». Un modèle du domaine, sous forme de diagramme de classes UML ou de diagramme Entité-Relations (E-R), représente graphiquement tout les objets (ou entités) du domaine et leurs relations. En d'autres termes, il définit le contexte en décrivant les objets du « monde réel ». On parle alors d'analyse orientée-objet, point de départ de la construction (ou programmation) orientée objets (Booch et al. 2007). Selon (Gordijn et al. 2000), le modèle du domaine (Business Model) définit « ce qui est offert, par qui, à qui, et ce qui est attendu en retour »[9].

[9] Il est important de distinguer le Business Model du (Business) Process Model (c.à.d. le modèle de processus métier), ce dernier définissant la dynamique du Business Model (le « comment ») pour comprendre comment sont orchestrés les services dans une entreprise (voir § 1.14.2).

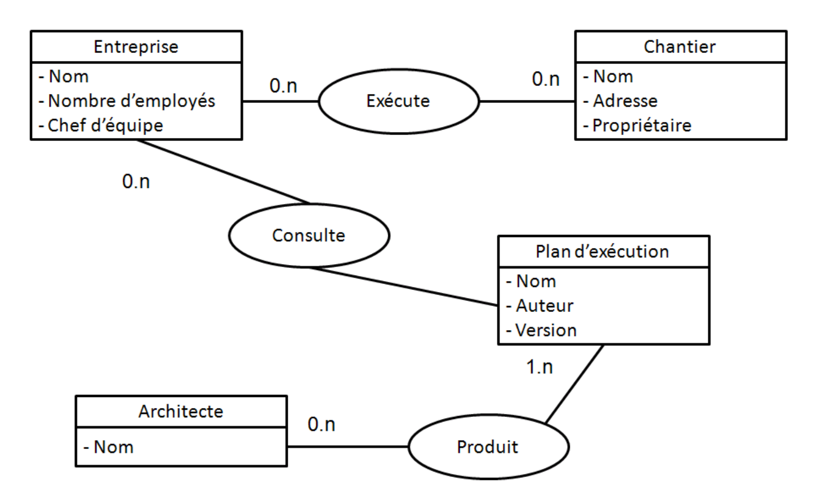



*Figure 20. Exemple de modèle métier sous forme de diagramme E-R*

N.B : Dans les projets de développement dédiés à des utilisateurs hors d'un contexte métier précis (ex. un guichet de retrait d'argent), il n'est pas nécessaire de mener cette activité.

<u>L'ingénierie des exigences (requirements engineering).</u>

L'ingénierie des exigences a pour but de guider la conception du logiciel. (Nuseibeh & Easterbrook 2000), cités dans (Sutcliffe 2007), la définissent pour les systèmes informatiques comme « le processus de découverte d'un objectif, en identifiant les clients et leurs besoins et en documentant ces informations dans un formulaire propice à l'analyse, la communication et à une ultérieure implémentation ». (Yeh & Zave 1980) identifiaient déjà il y a 30 ans que « la plupart des problèmes de développement de systèmes informatiques pourraient être assignés à une mauvaise compréhension ou spécification de ce que le système est supposé faire ». Ils soulevaient ainsi l'importance de traiter avec attention la phase de connaissance du besoin (ou phase d'expression des exigences) composée de :

- l'identification du besoin : il s'agit d'identifier et décrire les objectifs qui requièrent l'utilisation d'un système sous différentes formes (interviews, groupes de travail…),
- la compréhension du besoin : il s'agit de collecter et d'analyser de l'information à propos du système, de son environnement et des interactions entre les deux,
- la spécification du besoin : il faut alors décrire le comportement désiré du système[10].

On pourra rencontrer le terme « requirements elicitation » pour ce qui est de l'identification et la compréhension de l'existant (Loucopoulos & Karakostas 1995).

La formalisation des exigences (spécifications y comprises) se fait à travers un « Document d'Exigences Logicielles » (SRD pour Software Requirements Document). Le SRD exprime ce que le système doit faire mais pas comment, cette question étant relative à la conception. Il est le moyen de communication entre utilisateurs, experts, analystes et développeurs. De plus, les

---

[10] Selon (Jackson & Pamela Zave 1995), l'exigence « décrit une relation souhaitée parmi les phénomènes de l'environnement », puis la spécification décrit quant à elle « le comportement souhaité du système dans cet environnement ». Il nous précise que *l'exigence devient spécification* lorsque la relation souhaitée est partagée avec le système et contrôlée par ce dernier.

exigences ne doivent pas limiter la liberté de décision des concepteurs qui ont une connaissance plus approfondie des technologies et techniques d'implémentation (W. Davis 1999). Permettant la traçabilité des choix effectués en termes d'exigences, le SRD supporte également la validation par les utilisateurs (en phase exigences mais aussi en phase transition, c.à.d. après la conception).

Parmi les approches d'ingénierie des exigences, on pourra relever celles « orientées buts » (« Goal-Oriented Requirement Engineering ») (Cooper 1996; Kavakli 2002; Lamsweerde 2001; Lamsweerde 2003) et celles « basées sur les scénarios » (« scenario-based RE ») (Salinesi 2004; Sutcliffe 2003).

Les buts (goals) selon (Lamsweerde 2001), sont spécifiés afin de supporter l'expression et la validation des exigences ainsi que la gestion des conflits, la négociation, l'explication et l'évolution des exigences. En d'autres termes, identifier un but permettra de déterminer le besoin, (l'exigence) pour atteindre ce but. Le tableau suivant (Tableau 3), tiré de (Kavakli 2002), montre les différents apports de l'analyse des buts dans les activités d'ingénierie des exigences identifiées précédemment.

Tableau 3. Activités de RE et type de buts les supportant

| Activité de RE | Type de but | Apport |
|---|---|---|
| **Elicitation** | Buts courants | Compréhension de la situation organisationnelle existante |
| | Buts changeants | Compréhension du besoin de changement |
| **Spécification** | Buts futurs | Laison des buts métiers avec des composants système fonctionnels et non-fonctionnels |
| **Validation** | Buts d'évaluation | Validation de la conformité des spécifications du système avec les buts du client |

(Kavakli 2002) ajoute que « le processus d'ingénierie des exigences peut être vu comme une progression à travers quatre états de modélisation de la connaissance » décrits par ces quatre types de buts : l'état « tel quel », l'état « de changement », l'état « d'évaluation » et l'état « à être ». Il précise que le cheminement à travers ces quatre états n'est pas figé et qu'il pourra varier en fonction du projet.

Il existe plusieurs méthodes d'ingénierie des exigences orientée buts (ex. KAOS, i*,NFR…) (Lapouchnian 2005) et les formalismes varient également[11]. De manière générale, lorsqu'on modélise des buts il faut pouvoir distinguer :

- les buts stratégiques (haut niveau d'abstraction) relatifs aux objectifs du domaine (ex. « assurer la prise en charge des patients), et les buts opérationnels (bas niveau d'abstraction) plus spécifiques (ex. « signaler la prise en charge du patient sur le tableau). Il peut exister plusieurs niveaux de buts stratégiques et opérationnels. On dira qu'un but contribue à l'atteinte du but qui lui est supérieur (voir figure 6).

[11] http://www.utdallas.edu/~supakkul/tools/RE-Tools/

- les buts fonctionnels (ce qu'il faut faire) et les buts non fonctionnels qui sont des qualités supplémentaires souhaitées (sécurité, coûts, usabilité…),
- les buts « génératifs » (atteindre/cesser), « restrictifs » (maintenir/éviter) et les « softgoals » qui sont relatifs à des préférences et évalués de manière qualitative et subjective contrairement aux deux autres types,
- les relations entre les buts : ET (plusieurs buts contribuent à l'atteinte du but supérieur), ET complet (plusieurs buts contribuent et suffisent à l'atteinte du but supérieur), OU (le but inférieur est une alternative au but supérieur).

La figure suivante (Figure 21) est un exemple de diagramme de buts fonctionnels dans le domaine du projet AIC. Elle illustre un besoin courant mais crucial qui est d'assurer la bonne exécution du chantier en faisant en sorte que les entreprises de construction accèdent aux documents à exécuter tout en évitant qu'ils n'utilisent des documents rendus obsolètes par des mises à jour.

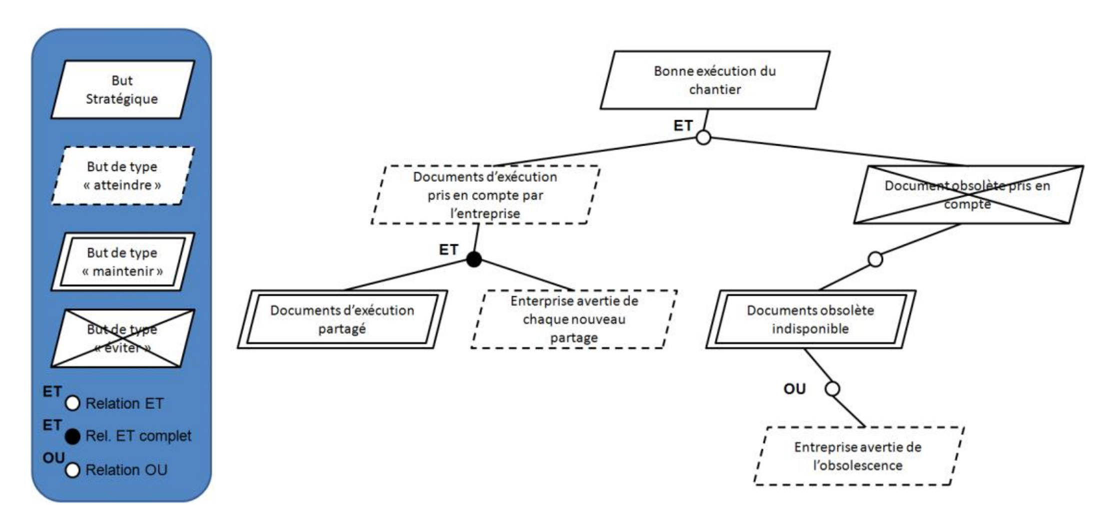



*Figure 21. Exemple de diagramme de buts*

Les scénarios sont la représentation du monde réel. Les méthodes basées sur les scénarios complètent les buts en rendant les intentions plus claires à travers des descriptions de fonctionnement qui satisfassent ces buts. ScenIC et SCRAM en sont deux exemples (Misra & Kumar 2005). On retrouve dans les typologies de scénarios un découpage similaire à celui observé précédemment pour les buts (voir tableau 1), à savoir :

- les scénarios de déclaration du problème qui expliquent en quoi le système actuel n'est pas satisfaisant,
- les scénarios de vision décrivent comment le système devrait opérer,
- les scénarios d'usage (ou comportementaux) décrivent les comportements des utilisateurs et du système actuel, ainsi que leur contexte physique et social. Ils sont aussi utilisés comme donnée de test pour la validation des exigences.

Les buts et les scénarios servent de base à la modélisation du système, tantôt transformés en modèles conceptuels, tantôt utilisés comme source d'inspiration pour le développement via prototypage (voir section suivante). Étant issus des témoignages des utilisateurs, ils présentent des limites relatives à la subjectivité (oubli d'information ou au contraire exagération des problèmes, différences d'opinions…).

Alors que les buts traduisent des intentions peut être trop vagues et que les scénarios paraissent au contraire trop spécifiques, il semblerait pertinent selon (Misra & Kumar 2005) de combiner ces deux approches qui s'avèrent complémentaires.

Les activités de l'analyse et de la conception (analysis and design).

L'analyse est définie comme le fait de poser et comprendre le problème et son domaine en faisant abstraction de l'implémentation. Les inputs des activités d'analyse sont donc le modèle du domaine et les exigences que l'on aura définies précédemment. Elles définissent elles-mêmes les inputs nécessaires à la conception au travers de plusieurs modèles créés à l'aide d'outils dédiés :

- un diagramme UML de cas d'utilisation (use case diagram) (Alistair Cockburn 2000) décrivant de manière graphique un ensemble d'évènements ayant lieu entre un ou plusieurs acteurs et le système pour accomplir un but précis. Notez que le diagramme de cas d'utilisation peut aussi être utilisé pour modéliser des buts (on parle alors de « business use case »).

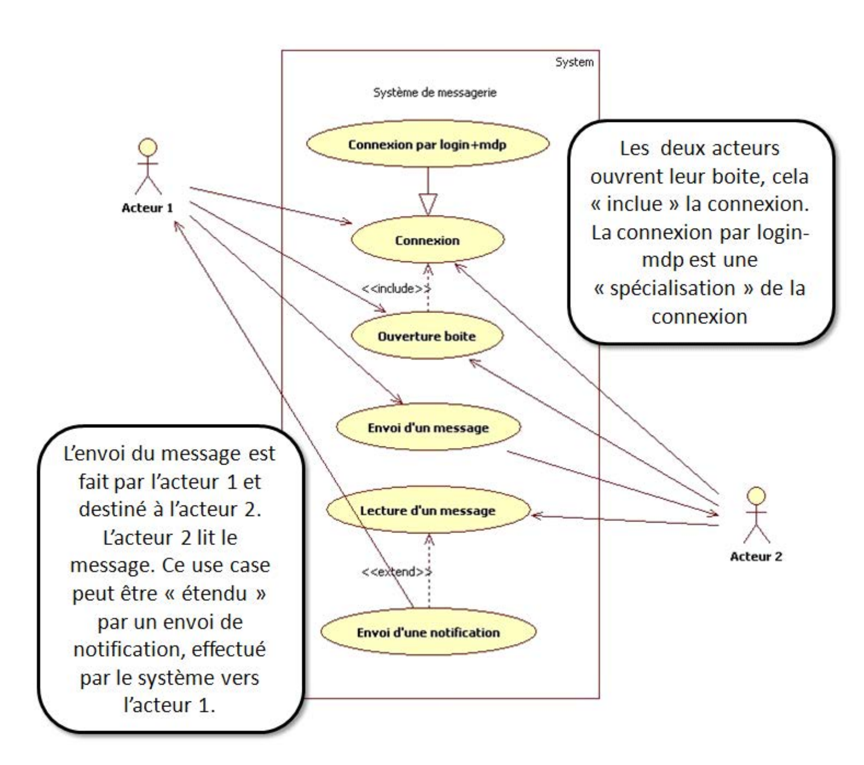



*Figure 22. Exemple de diagramme UML de cas d'utilisation*

- Un diagramme UML de séquences[12] décomposant chaque use case, en montrant les échanges de messages entre un acteur et le système ainsi que des indications en cas d'itération. Le système est alors vu comme une « boite noire », sans détail de ce qui le compose.

[12] http://www.agilemodeling.com/artifacts/sequenceDiagram.htm

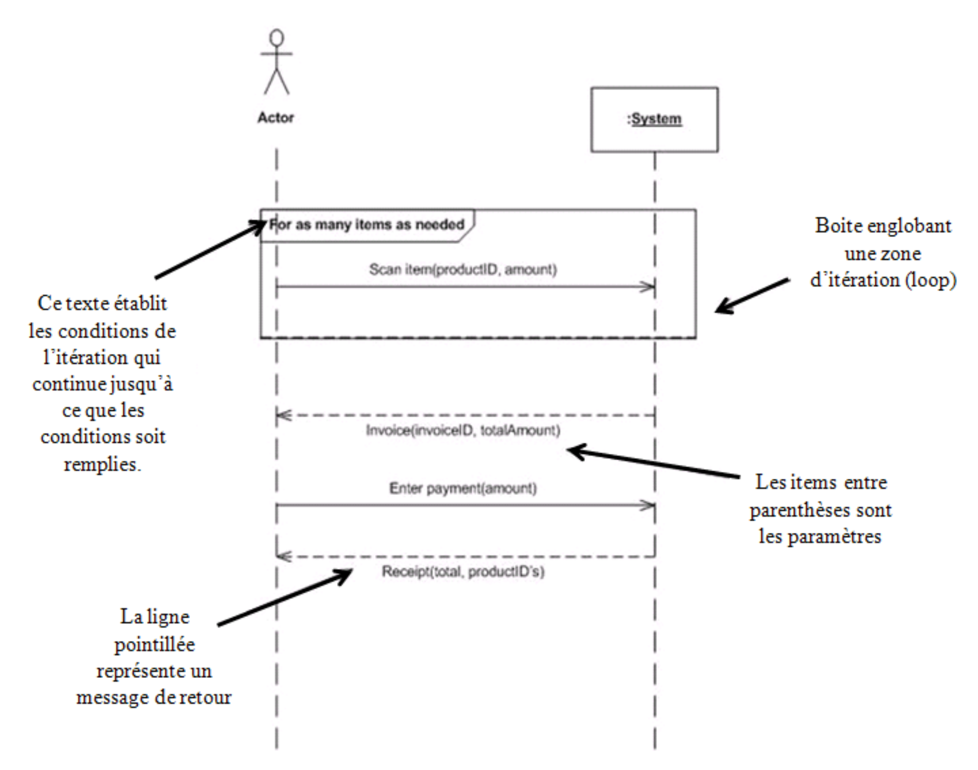



*Figure 23. Exemple de diagramme de séquence système*

Alors que l'analyse décrit le 'quoi faire' (après avoir défini le 'pourquoi ' avec les exigences), la conception définit le 'comment faire'. Cette activité implique l'interprétation du concepteur, c'est l'activité créative du processus et la plus subjective. Elle consiste à interpréter l'analyse pour proposer une solution dans une architecture logicielle et matérielle à travers des classes et des abstractions d'objet : c'est ce qu'on appelle l'architecture conceptuelle (Bass et al. 2003). Elle représente à la fois l'ensemble des évènements et l'ensemble des données (Guibert 2007).

- Le concepteur étend le diagramme de séquences système en un diagramme de séquences technique dans lequel il y ajoute les évènements internes au système qui est alors décrit plus précisément (sous forme d'entités logicielles ou de composants, il est qualifié de « boîte blanche »). On parle ici de « décomposition fonctionnelle ».

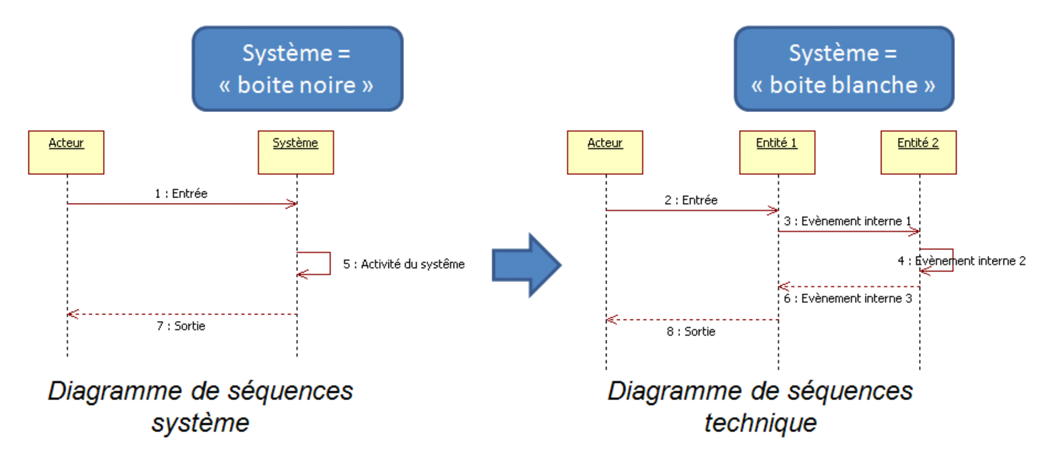



*Figure 24. Représentation des évènements internes au système par un diagramme de séquence technique*

- Il transforme le modèle conceptuel du domaine en un (autre) modèle relationnel (E-R ou de classes UML) qui modélise quant à lui toutes les informations (et leurs structures) devant être

manipulées (et donc stockées) afin de connaitre leur nature et donc le moyen de les traiter. Les structures de données sont typées : texte, lien hypertexte, donnée numérique, date/heure, objet multimédia, valeur booléenne. C'est ce qu'on appelle « l'organisation structurelle ». Cette structure sera utilisée par le développeur pour implémenter la solution dans son environnement de développement spécifique. Plus cette structure contient d'éléments déjà définis et codés, plus le développeur pourra gagner de temps en réutilisant ce code.

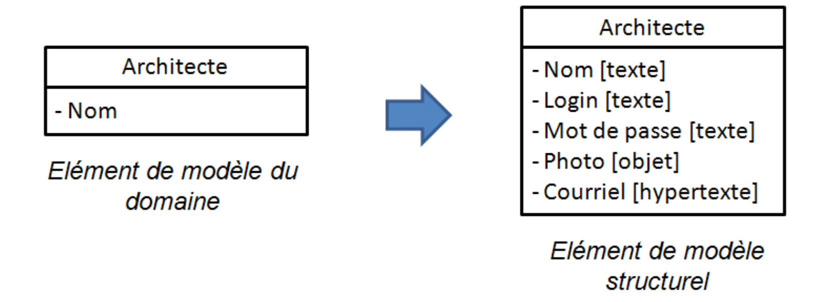



*Figure 25. Exemple de représentation d'un concept métier en élément de modèle structurel*

Les activités de l'implémentation.

L'implémentation d'un logiciel est l'écriture du code interprété par le dispositif qui l'exécute. C'est donc originellement une saisie de texte, dans un langage particulier et via un éditeur. Au travers des générations, les langages de programmation (O'Regan 2008) se sont éloignés du « langage machine » (fait de 1 et de 0) pour devenir plus conceptuels et ainsi être plus facilement manipulables et compréhensibles par les humains. On parle de langages « haut-niveau » qui peuvent être créés indépendamment de la machine à partir de librairies puis compilés pour différentes plateformes. Les langages orientés objet en font partie et ont la particularité de diviser les tâches de programmation en objet (ex. Java, C++…). UML propose de modéliser le système sous forme de diagrammes de composants (qui permet de montrer les composants du système d'un point de vue physique, tel qu'ils sont mis en œuvre).

Il existe plusieurs modèles qui décrivent une architecture logicielle, introduisant différents concepts (Bass et al. 2003). Le modèle MVC (Modèle-Vue-Contrôleur) offre un point de vue simple au travers de trois classes (Figure 26).

- Le Modèle contient les données mais aussi les méthodes pour manipuler ces données (c.à.d. ajouter, supprimer, récupérer…). C'est en quelque sorte le « cœur » de l'application.
- La Vue est ce avec quoi l'utilisateur interagit. Symboliquement parlant, une vue est une fenêtre sur le modèle : elle présente une partie du contenu de celui-ci. La vue a aussi pour rôle de capter les interactions de l'utilisateur (par exemple les clicks).
- Le Contrôleur synchronise la vue et le modèle en les mettant à jour. Typiquement, il analyse les requêtes envoyées par l'utilisateur depuis la vue (ex. afficher des données particulières comme l'emploi du temps d'un jour particulier), il demande au modèle approprié d'exécuter la requête (ex. au modèle d'emplois de temps de récupérer celui du jour demandé) puis à la vue de s'actualiser avec les informations envoyées par le modèle.

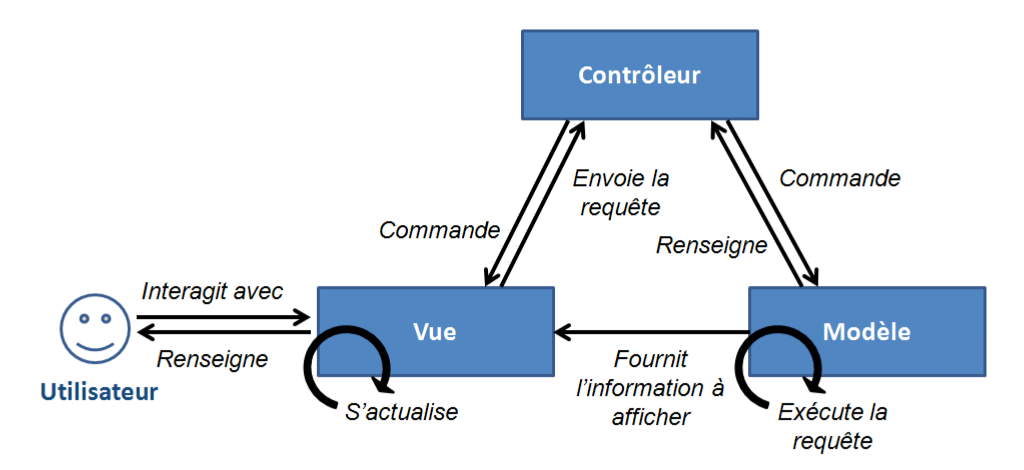



*Figure 26. Schéma d'une architecture logicielle représentée par le modèle MVC*

Le kit de développement logiciel (SDK pour « software development kit ») est l'ensemble des outils de développement qui seront fournis au développeur et permettront la création du logiciel (codage et compilation) dans un environnement matériel et logiciel spécifique. Les SDKs contiennent également des notes techniques et autres documentations pour assister le développeur.

<u>Les activités de test et le déploiement.</u>

Un cycle de développement (qu'il soit global ou qu'il fasse partie d'un processus itératif) est évalué à travers l'évaluation de ses différentes phases (Kaner & Falk 1999).

Durant les phases de planification (c.à.d. d'études préliminaires et d'élaboration), il est nécessaire d'évaluer les exigences (de par l'analyse du SRD) et les définitions fonctionnelles apportées par les activités de conception.

- Les exigences doivent être estimées cohérentes, vis-à-vis des besoins identifiés (le produit « exigé » est-il utile ? Pendant combien de temps le sera-t-il ?) mais aussi des possibilités de développement (quels sont les moyens personnels, techniques, financiers à disposition ? Est-ce raisonnable d'entamer le développement?).
- Le produit conçu (et proposé) doit répondre aux exigences, à savoir aux besoins mais cela signifie aussi qu'il doit être réalisable techniquement (en fonction des performances du dispositif qui l'exécutera).

L'évaluation de la solution implémentée se fait au travers de tests d'utilisation, exécutés en interne dans l'équipe de développement (identification des bugs majeurs) puis auprès des clients qui évalueront son utilité (test de l'acceptation). Le produit ne peut être testé qu'à partir du moment où il est (en partie) fonctionnel.

Lors du déploiement, c'est-à-dire lors de la livraison du produit, il peut être nécessaire d'en assister l'installation. Le produit doit ensuite être « maintenu » à travers l'assistance technique, la correction de bugs persistants éventuels voire l'évolution de la solution (nouvelles fonctionnalités, changement de l'interface…).

### 1.9.4 *Les méthodes agiles*

Au-delà des activités identifiées, l'activité de gestion du projet occupe une place très importante. Comme dans tout travail collaboratif, il est crucial de (1) pouvoir coordonner et superviser les tâches de chacun et (2) assurer la communication et le transfert d'informations entre les différents acteurs de l'équipe et à tous les stades du développement. À la fin du 20e siècle, nombre de chercheurs estimèrent que les cycles de développement traditionnel ne répondaient pas à ces besoins et proposèrent leurs propres méthodes visant à améliorer le développement logiciel. Afin de les unifier, ils instaurèrent les principes fondamentaux d'un développement adapté des logiciels : le développement agile.

Les principes

Le manifeste pour le développement agile de logiciels[13] (Greer & Hamon 2011; Martin 2003) rédigé en 2001 définit 12 principes déclinés des quatre valeurs fondamentales suivantes (voir aussi Figure 27) :

- la priorité des individus et des interactions sur les procédures et les outils, en encourageant l'auto-organisation et la motivation,
- la priorité aux logiciels opérationnels qui seront plus utiles aux clients pour évaluer le travail en cours (et donc mieux accueillis) que des documentations exhaustives,
- la priorité à la collaboration avec les clients plus qu'à la négociation contractuelle avec eux, impliquant ceux-ci tout le long du cycle de développement afin d'assurer la prise en compte continue de leurs exigences (et plus uniquement au début du projet pour la définition du contrat),
- la priorité à l'adaptation au changement plus qu'au suivi d'un plan, autorisant l'évolution du processus.

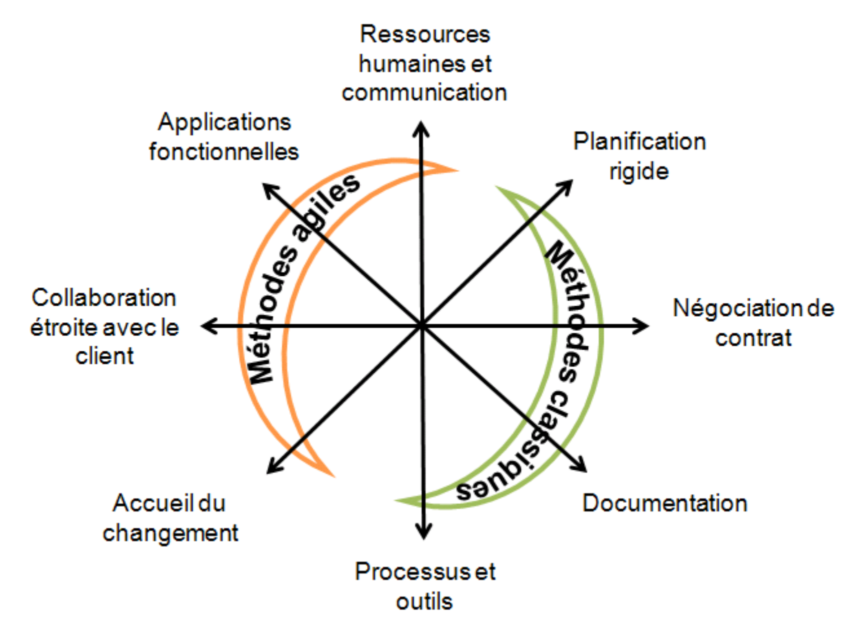



*Figure 27. Valeurs des méthodes agiles face aux concepts des méthodes classiques*

[13] http://www.agilemanifesto.org/iso/fr/

Plusieurs méthodes intègrent ces notions à leur manière, deux d'entre elles sont décrites dans la section qui suit : XP et SCRUM.

La méthode extreme programming (XP)

Parmi les méthodes de développement agile, l'XP (Beck 2006) « pousse à l'extrême » les bonnes pratiques de développement classique. Cette méthode préconise des cycles de développement très courts (de 2 à 3 semaines) basés sur des « user stories » fournis par le client. Alors que le use case décrit l'utilisation du système de manière générale et formalise de façon permanente l'accord entre le client et le développeur, les « user stories » (« histoires d'utilisateurs ») supportent une collaboration continue entre ces deux acteurs en décrivant des scénarios d'utilisation précis. Chaque « user story » est éphémère et faite pour être oubliée dès lors que la solution a été implémentée et validée (Cohn 2003). L'utilisateur est ainsi fortement impliqué dans le processus.

En fonction des « user stories » et itérativement jusqu'à ce que tous les cas de figure soient traités et validés, les développeurs suivent le processus suivant :

- ils déterminent et attribuent les tâches de développement ainsi que les tests à mener,
- ils réalisent leurs tâches de développement en binôme,
- ils testent le code développé et l'intègrent continuellement dans le système final.

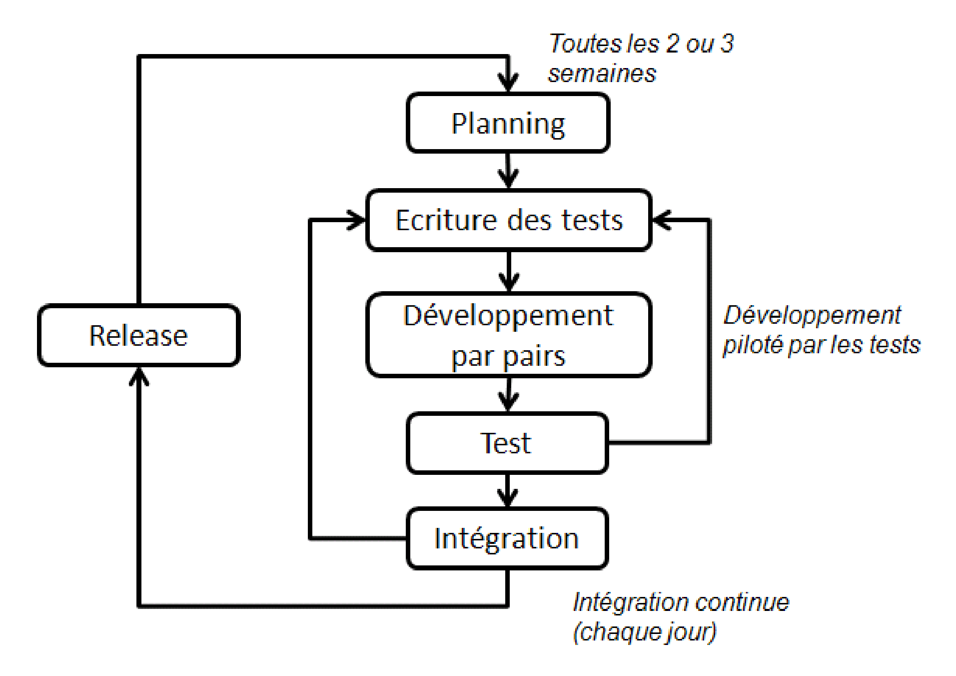



*Figure 28. Cycle de développement dans la méthode XP*

Cette méthode ne s'inscrit pas du tout dans la méthodologie présentée précédemment car elle est clairement orientée programmation (pas de phase de conception) et n'est pas supportée par des modèles. Elle présente cependant l'avantage de gérer continuellement l'évolution des besoins et des risques. Le fonctionnement par petites équipes et l'autonomie des binômes de développeurs permet aussi de simplifier l'activité de gestion du projet.

La méthode SCRUM

Le nom « scrum » se rapporte au domaine du rugby : c'est cette manière de recommencer le jeu répétitivement tout au long d'une partie. La méthode de développement agile SCRUM (Schwaber & Sutherland 2011; Schwaber & Beedle 2001) est également basée sur le fractionnement du développement en cycles courts (1 à 4 semaines) qui sont ici nommés les « sprints ». Les phases de « backlog » servent à déterminer les exigences (sous forme de « user stories ») pour le produit en général puis plus précisément pour le sprint qui suit. La particularité de la méthode SCRUM est qu'une fois le sprint commencé, il n'est pas interruptible. L'équipe de développement travaille alors sans contact avec le client et s'organise elle-même. Les exigences restent donc inchangées durant chaque cycle et c'est le « ScrumMaster » qui s'assure du bon déroulement du développement et que l'équipe n'est pas interrompue. Au cours du sprint, des réunions journalières (les mêlées) servent à faire le point sur l'avancement. A la fin de chaque sprint, une « démo » de ce qui a été produit est présentée et l'équipe effectue une rétrospective de ce qui a été fait (et pas fait).

Ici encore, la gestion du projet est une considération importante. Celui-ci est géré par des cycles courts conclus par la mise en commun du travil et des discussions entre développeurs et utilisateurs.

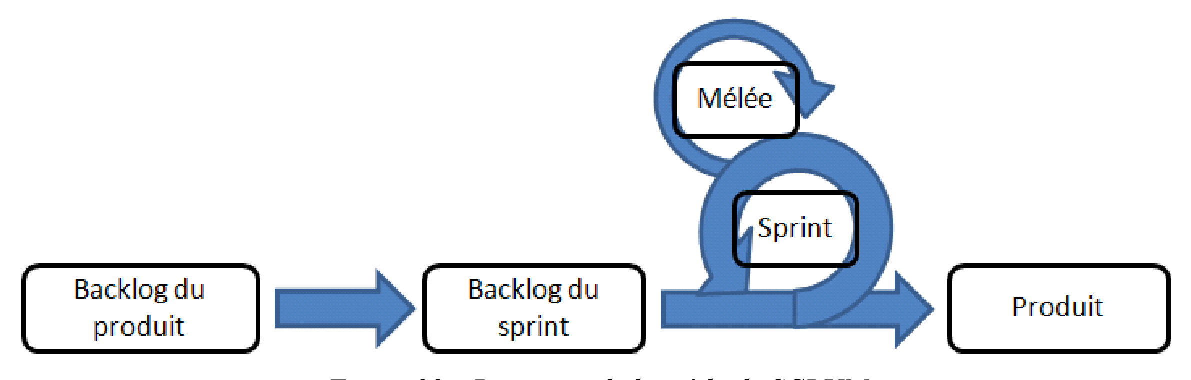



*Figure 29. Processus de la méthode SCRUM*

### *1.9.5 Constat et conclusion*

Le projet de développement logiciel suit un cycle de vie structuré et formalisé.. Les méthodes modernes s'appuient sur des modèles qui rendent compte des activités menées et servent de point de départ aux activités qui suivent, ce qui assure la cohérence des résultats tout au long du processus mais aussi la traçabilité des choix. Les formalismes utilisés sont admis par la communauté et sont compris par chacun.

L'agilité apporte aux méthodes de développement une organisation rigoureuse des cycles de développement. L'alternance de cycles courts « exigences-développement-bilan » permet de construire une solution logicielle petit à petit et toujours dans le respect des attentes du client. Ce dernier peut exprimer ses besoins de manière simple à travers des modèles adaptés (principalement les user stories) et les développeurs peuvent alors s'auto-organiser pour produire une solution fonctionnelle qui sera rapidement évaluée. Ces méthodes assurent alors l'utilité du produit final proposé.

Ce que l'agilité n'apporte pas et qui reste la lacune des méthodes de génie logiciel, c'est la prise en compte des utilisateurs et de leurs interactions avec le système (Constantine 2002). En effet, pour une bonne appropriation d'un outil par un utilisateur, il est nécessaire d'en définir non seulement le fond (c.à.d. les fonctionnalités) mais au aussi la forme, l'interface : on parle alors d'interface homme-machine (IHM) (Booth 1989). Les méthodes agiles n'offrent cependant pas de réels guides pour les concepteurs et les développeurs en ce qui concerne les IHM. Les méthodes dites « enrichies » apportent cette composante interface en orientant le développement sur « l'utilisabilité ».

## 1.10 La conception d'IHM : de l'utilisateur à l'interface

Si les méthodes classiques de Génie Logiciel sont basées sur l'étude de « ce que le logiciel fait » les méthodes enrichies de conception d'IHM s'intéressent en plus à « comment il doit le faire ». Cette question jusque-là laissée au libre choix du développeur pendant le développement est désormais traitée dès l'analyse et la conception à travers plusieurs approches : la conception centrée utilisateur (CCU), la conception centrée activité (CCA), la conception dirigée par les buts (CDB) et la conception centrée usage (CCUs). Cette section illustre comment ces approches combinent études de terrain et implications diverses de l'utilisateur dans le processus de conception. Nous y présentons plusieurs types de modèles (caractérisant les tâches utilisateurs et les concepts qui gravitent autour) qui sont utilisés pour la conception.

### *1.10.1 Vers une méthodologie de la conception enrichie*

Les méthodes de GL et notamment les méthodes agiles visent la satisfaction du client, considéré comme celui qui « a besoin » du logiciel. Le client est alors impliqué dans la formulation des exigences puis dans l'évaluation de la réponse à ces exigences. Nous considérons à présent le client comme le futur utilisateur. En d'autres termes, il ne s'intéresse plus seulement à ce que va lui apporter le logiciel (c.à.d. à la manière dont il répondra à ses besoins), mais il attend en plus une utilisation simple et efficace à travers une interface adaptée : on parle d'utilisabilité. La capacité d'une IHM à s'adapter aux contraintes à la fois matérielles et environnementales dans le respect de son utilisabilité est définie comme sa plasticité (Thevenin 2001).

(Laurillau 2002) illustre l'enrichissement des méthodes de conception logicielle à travers l'exemple de la méthode « en V » (Figure 30). Il réorganise les étapes de ce cycle « selon deux catégories : les étapes relevant de l'espace IHM, c'est-à-dire les étapes de conception ergonomique du système et d'élaboration d'un modèle d'interaction répondant à des requis identifiés à l'issue de l'étape d'analyse des besoins, et les étapes relevant de l'espace logiciel, c'est-à-dire les étapes de conception et de développement logiciels ».

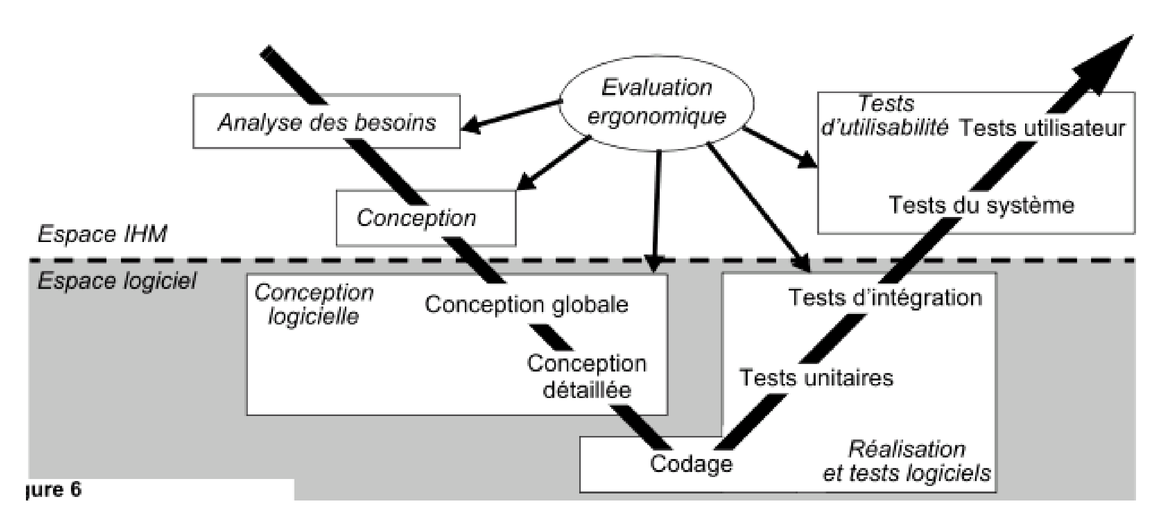



*Figure 30. Le processus de conception « en V », repris par (Laurillau 2002)*

Plusieurs approches visent à préconiser l'utilisabilité:

- La conception centrée utilisateur (Norman 1986)
- La conception centrée activité (Gifford & Enyedy 1999)
- La conception dirigée par les buts (Cooper 1996)
- La conception centrée usage (Constantine & Windl 2003)

Littéralement parlant, la CCU se concentre sur l'utilisateur (qui manipule le système), la CCA sur l'activité (c.à.d. les tâches qui doivent être proposées par le système) et la CDB sur les buts (liés à l'utilisation de système). Elles semblent donc être des approches différentes. Pourtant, la définition de l'utilisateur dans les méthodes de CCU s'est étendue de sorte qu'elle couvre aussi l'identification des ses motivations et de ses tâches. Réciproquement, l'analyse des tâches d'un système (dans la CCA) est liée aux tâches de l'utilisateur et donc à la connaissance de celui-ci. Il en est de même pour l'analyse des buts dans la CDB. Ces conclusions sont notamment celles de l'étude menée dans (Williams 2009), qui souligne que la limite qui sépare ces approches est floue et difficile à déterminer.

Constantine, à l'origine de la conception centrée usage, se base aussi sur l'analyse de l'utilisateur, de ses tâches et objectifs. Il cherche cependant à limiter l'intervention de l'utilisateur en donnant la priorité à l'abstraction par les modèles sur (1) les données brutes à propos des utilisateurs et (2) les prototypages réalistes et itératifs (Constantine & Windl 2003).

Tableau 4. Différences majeures entre CC-Utilisateur et CC-Usage (Constantine, 2003)

| Conception centrée utilisateur | Conception centrée usage |
|---|---|
| Concentration sur l'utilisateur, son expérience, sa satisfaction | Concentration sur l'usage et sur l'outillage des tâches |
| Dirigée par les données sur l'utilisateur | Dirigée par les modèles |
| Implication importante de l'utilisateur | Implication sélective de l'utilisateur |
| Description des utilisateurs, de leurs caractéristiques | Modèles de relations entre l'utilisateur et le système |
| Maquettages réalistes | Maquettages abstraits |
| Conception par prototypage itératif | Conception par modélisation |
| Processus variés et souvent informels | Processus systématiques et entièrement spécifiés |
| Évolution à travers la résolution d'erreurs | Évolution à travers l'ingénierie |

Ici encore, notre objectif n'est pas de décrire les méthodes pour ce qu'elles sont mais plutôt de comprendre les concepts qui caractérisent la conception. Nous préfèrerons parler plus généralement de la méthodologie de la conception enrichie en la définissant au travers de ses méthodes. Nous adoptons la définition de (Abras & Maloney-Krichmar 2004) (qu'il propose originellement pour la conception centrée utilisateur) : **« un ensemble de processus dans lesquels les utilisateurs finaux influencent - d'une façon ou d'une autre - la mise en forme de la conception »**.

Ainsi, selon (Norman 1986), l'utilisateur d'une interface adaptée devra:

- savoir ce qu'il doit faire en fonction de ce qu'il veut faire (ex. si il veut retourner à l'accueil d'un site, si il veut valider et payer sa commande…). Cela implique que les fonctionnalités soient visibles et compréhensibles : on parlera alors d'affordance[14] (Maier, 2009). Le concepteur pourra également utiliser les contraintes pour limiter les choix de l'utilisateur et le guider vers la solution.
- Savoir ce qu'il se passe et ce qu'il va se passer (ex. « le fichier est en cours de téléchargement », ou « si je clique sur annuler la transaction s'arrêtera »).

Il est préconisé d'utiliser des représentations graphiques simples plutôt que des mots ainsi que de simplifier la structure des tâches afin de limiter le recours de l'utilisateur à sa mémoire. Toute erreur dans un processus de tâches doit pouvoir être corrigée. Un système dit « centré utilisateur » ne nécessite pas de manuel d'utilisation très étoffé : il se veut intuitif.

Le tableau suivant (Tableau 5) présente quelques techniques et objectifs relatifs à l'implication du client au travers des phases d'un cycle de projet de développement logiciel. Sont signalés en rouge les apports des méthodes enrichies par rapport aux méthodes traditionnelles de Génie Logiciel (GL).

[14] Le terme affordance vient du verbe « to afford » (offrir, fournir, procurer) est désigne la capacité d'un objet à suggérer sa propre utilisation.

Tableau 5. Implication de l'acteur en tant que client et utilisateur

| Technique | But | Phase |
|---|---|---|
| **Groupe de travail** | Connaitre les exigences du client | Études préliminaires |
| **Groupe de travail / observations** | Connaitre le domaine d'utilisation | Études préliminaires |
| **Groupe de travail / observations** | Connaitre les tâches métier qu'il faut accomplir | Études préliminaires |
| **Interviews et questionnaires** | Connaitre l'utilisateur et ses préférences | Études préliminaires |
| **Observations sur site** | Connaitre le contexte physique d'utilisation | Études préliminaires |
| **Tests, simulations** | Évaluer les alternatives d'interaction + obtenir des informations supplémentaires sur les préférences | Élaboration/ Construction |
| **Tests et questionnaires** | Évaluer l'utilité | Transition |
| **Tests et questionnaires** | Évaluer l'utilisabilité | Transition |

### *1.10.2 La modélisation de l'utilisateur et de son contexte*

Selon (Dey 2001), « le contexte regroupe toutes les informations qui peuvent être utilisées pour caractériser la situation d'une entité. Une entité est une personne, un endroit ou un objet qui relève de l'interaction entre un utilisateur et une application, eux-mêmes inclus ».

Le modèle contextuel de (Tazari 2003) contient par exemple : le profil de ressources matérielles et logicielles, le profil de localisation (c'est-à-dire la position de l'utilisateur), l'instant présent (heure, date courante), le profil de l'utilisateur (expert, confirmé, novice) dans l'activité, les préférences sur les applications (paramétrage de la position des menus,…), données spécifiques aux applications, les buts de l'utilisateur… (Calvary et al. 2003) regroupe ces informations à travers trois concepts majeurs : l'utilisateur, la plateforme (logicielle et matérielle) et l'environnement.

Dans le processus de conception des IHM, la prise en compte d'un ou plusieurs élément(s) du contexte peut être déterminante tout comme elle peut être inutile. Il est donc important de savoir déterminer quel rôle joue le contexte et comment le prendre en compte.

En 1999, Cooper introduisait la CDB et une modélisation innovante de l'utilisateur et son contexte dans le domaine des IHM : le persona (Cooper 1999). « Le persona est une description précise des caractéristiques d'un utilisateur et de ce qu'il veut accomplir ». Il peut représenter un utilisateur particulier ou un ensemble d'utilisateurs, un profil (Chang et al. 2008). On apprend également de (Chang et al. 2008) que le persona est créé en phase d'études puis complété si de nouvelles informations sont acquises lors des tests. Il peut éventuellement être imaginé à l'issue

d'une idée de conception particulièrement innovante, auquel cas l'utilisateur type n'existe pas encore.

Le persona a aussi été adopté dans les méthodes de CCU. En fonction des cas d'études, il arbore différentes formes et différents contenus (Pruitt & Grudin 2003; Olsen 2004; Seffah et al. 2009). De manière générale, la description de l'utilisateur au travers d'un persona doit apporter toute information qui pourra s'avérer utile dans la conception du logiciel qu'il devra appréhender (ex. les principales tâches de maintenance d'un méthodiste qui sont utilisées pour concevoir un système de e-maintenance industrielle au niveau d'un groupe agro-alimentaire (Seffah et al. 2009)). Certains feront le choix de « tout dire » puis de piocher l'information nécessaire dans ces données, d'autres limiteront la description à ce qui sera utile en omettant les informations superflues. Cette information pourra être d'ordre personnel comme professionnel, et définir via différents attributs une identité, une activité, une expérience, une motivation, un contexte physique, un contexte social, etc…

Constantine dans son approche de conception centrée usage utilise des « rôles utilisateur » (user role) pour « informer le processus de conception » à propos de l'utilisateur (Constantine 2006). Le rôle utilisateur est défini comme « un ensemble de besoins caractéristiques, d'intérêts, d'attentes, et de comportements en relation avec un système particulier ». Sa modélisation se fait par une « checklist » au travers de trois concepts principaux : le contexte, les caractéristiques, les critères (Figure 31).

En fonction des approches l'utilisateur arbore donc différentes caractérisations. La section suivante montre qu'il en va de même pour l'ensemble des tâches qu'il effectue en interaction avec le système.

Constantine & Lockwood, Ltd.

## User Role Checklist for Agile Modeling

A user role is a relationship with a system. Tasks are performed by users within roles. Tasks are about **what** users do, roles are about **how** they do it. The key to succinct characterization of user roles is **differential description**. How is this role not like other roles? What is distinctive or salient about it in comparison to other roles? Listed below are typical factors for consideration as potentially relevant and useful for characterizing a user role. They may or may not apply to a given role.

### Context (within which role is played)

**Is there anything special or distinguishing about this role in terms of:**

| | | Examples |
|---|---|---|
| ENVIRONMENT | ❑ overall job, workflow, or **activity** within which role is played | "Follow-up of prior purchase." |
| | ❑ **physical environment** in which role is played | "Typical noisy office." |
| | ❑ **social situation** in which role is played | "With field research partners." |
| | ❑ **relationships** with indirect users in role | "Customer on telephone." |
| | ❑ **external sources** of information, such as paper forms, telephone, visual observation, in-person interview | "Phone review of packing slip." |
| INCUMBENTS | ❑ **background** of role incumbents in terms of training, education, or experience | "Cursory OTJ training." |
| | ❑ **system knowledge** expected or required within role | "Fully familiar from long use." |
| | ❑ **domain knowledge** expected or required within role | "No retail management knowledge." |
| | ❑ distribution of **user skills** in terms of novice, intermediate, or expert usage patterns | "Mostly perpetual novices." |
| | ❑ **required** or discretionary nature of role | "Part of regular job." |

### Characteristics (of performance of role)

**Is there anything special or distinguishing about this role in terms of:**

| | |
|---|---|
| ❑ **orientation**, attitude, or emotional state typical within role | "Harried, under pressure." |
| ❑ **frequency** with which role is played | "Less than once a month." |
| ❑ **regularity** with which role is played | "Impulse buy after infomercial runs." |
| ❑ **intensity** of interaction in the role | "Sporadic bursts of calls." |
| ❑ **duration** of interaction in the role | "Full 8-hour shift." |
| ❑ **complexity** of interaction in the role | "No-brainer." |
| ❑ **predictability** of interaction in the role | "Scripted sales protocol." |
| ❑ **volume** of information handled in the role | "Limited items available." |
| ❑ **direction** of information flow to or from system | "Data entry." |

### Criteria (for support of role)

**Are there any design objectives that are particularly important for this role, such as:**

❑ ease of **learning**
❑ **retention** of learning
❑ **efficiency** of interaction
❑ **reliability** of interaction
❑ user **satisfaction**
❑ enhancement of **proficiency**
❑ user **convenience**
❑ **accuracy** of input
❑ **clarity** of presentation
❑ **comprehensibility** of presentation

**Are there any specific functions, features, facilities, capabilities, or content that are particularly important for this role to be performed effectively?**

Contact: lconstantine@foruse.com



*Figure 31. Modélisation d'un rôle utilisateur dans la conception centrée usage (tiré de (L. L. Constantine 2006))*

### *1.10.3 La modélisation des tâches et du contenu*

Nous adoptons le point de vue de (Kaindl & Jezek 2002), qui définit la tâche « comme un fragment d'activité qu'une personne doit effectuer », où l'activité est ici l'interaction avec un système.

Il existe plusieurs niveaux d'abstraction de tâches : de l'intention guidée par une réflexion (on pourra parler de tâche cognitive) à l'interaction physique proprement dite. (Winckler et al. 2004) en donne par exemple 4 niveaux (voir Tableau 6).

Tableau 6. Niveau et portée des tâches selon [Winckler2004]

| Niveau | Portée |
|---|---|
| **But** | Buts et intentions de l'utilisateur |
| **Abstrait** | Décomposition des buts en tâches génériques |
| **Interaction** | Actions de l'utilisateur et du système |
| **Présentation visuelle** | Rendu des fonctions exécutées par l'application |

(Samaan 2006) prend en compte deux niveaux pour le modèle de tâches. « Le modèle abstrait de tâches représente une vue générique du système modélisé, et cela indépendamment du contexte d'utilisation en général et plus spécifiquement indépendamment de la plate-forme ». Une tâche dite « abstraite » peut alors suivre plusieurs « patterns d'interaction », c'est-à-dire qu'elle peut être réalisée de plusieurs façons à travers des ensembles de tâches « du modèle concret », comme par exemple, sélectionner un élément avec la souris ou avec le clavier... Ainsi, dans le modèle concret, « les tâches représentent des interactions concrètes avec des objets de la présentation manipulée ».

Le contenu d'une interface est composé d'objets d'interaction qui peuvent également être spécifiés de manière concrète (CIO pour Concrete Interaction Object) ou abstraite (AIO pour Abstract Interaction Object). Selon (Bodart & Vanderdonckt 1996), le CIO « représente tout objet d'interface visible et manipulable qui peut être utilisé en entrée ou en sortie d'information relative à une tâche d'interaction ». Le manque d'uniformisation et de standardisation dans les CIO limite la compatibilité avec la programmation orientée objets qui manipule des concepts abstraits. Cela les rend difficiles à réutiliser et à maintenir.




*Figure 32. CIO similaires dans différents environnements*

Un AIO « consiste en un ensemble de CIOs d'une même catégorie mais définis indépendamment de leur environnement d'implémentation ». La manipulation d'objets abstraits permet de définir une interface qui sera réutilisable. Elle permet aussi de laisser libres les développeurs quant à l'implémentation de la solution en fonction de l'environnement. Enfin, les

AIO sont la clé de la génération automatique d'interfaces dirigée par les modèles (voir la section suivante (1.11)). (Bodart & Vanderdonckt 1996) répartissent les AIO en six familles : les objets d'action, de défilement, statiques, de contrôle, de dialogue et de réaction (feedback).

Tableau 7. L'ensemble des AIO répartis en 6 familles (extrait de (Bodart & Vanderdonckt 1996))

| AIO Sub-class | Sub-class Members |
|---|---|
| Action AIO | menu, menu item, menu bar, pull-down menu, pop-up menu, cascading menu, submenu, embedded menu, removable menu, expandable menu, compound menu |
| Scroll AIO | scroll arrow, scroll cursor, scroll bar, frame |
| Static OIA | label, separator, group box, prompt, static icon |
| Control AIO | single line edit box, multi-line edit box, profiled edit box, command box, scale, thermometer, gauge, jogger, check box, switch, option box, radio button, radio icon, dynamic icon, spin button, push button, drawn button, cyclic button, 2D and 3D tool boxes, list box, unitary list box, Boolean list box, file selection list box, drop-down list box, scrolling list box, combination box, drop-down combination box, scrolling combination box, normal table, extended table, value set |
| Dialogue AIO | window, help window, logo window, edit windows, dialogue box, expandable dialogue box, repetitive dialogue box, radio dialogue box, panel, control panel |
| Feedback OIA | information message, warning message, help message, error message, progression indicator, contextual cursor |

On distingue ces niveaux d'abstraction dans les différentes méthodes de modélisation. Les paragraphes suivants illustrent comment le scénario et le use case utilisés dans les activités de développement logiciel peuvent supporter la caractérisation des interactions avec plusieurs niveaux d'abstraction. Nous y présentons ensuite un type de modélisation spécifique et plus adapté : les arbres de tâches. Enfin, nous montrons comment illustrer une interface à travers des maquettages, ici aussi de façon concrète ou abstraite.

Les modélisations du GL

La modélisation par scénario, que nous avons introduite avec l'activité d'ingénierie des exigences (voir § 4.1.2) est utilisé dans les approches CCU, CCA et CDB pour décrire le comportement d'un persona. D'après (Cohn 2003) il se veut plus précis qu'une « user story ». Comme on peut le voir dans l'exemple ci-dessous (inspiré de l'exemple de (Cohn 2003)), un scénario peut décrire littéralement des intentions (« veut y planifier un voyage ») des tâches abstraites (« accède à notre site »), des tâches d'interaction (« click », « saisit ») et des représentations visuelles (« un formulaire de saisie »).

> *« Amy est intéressée par la culture japonaise et veut y planifier un voyage. Elle accède à notre site internet et click sur le lien Hôtels. Un formulaire de saisie apparaît. Elle saisit Nagoya comme ville de séjour et les dates du 14 au 19 novembre. Elle sélectionne parmi une liste d'hôtels le Royal Park Inn ».*

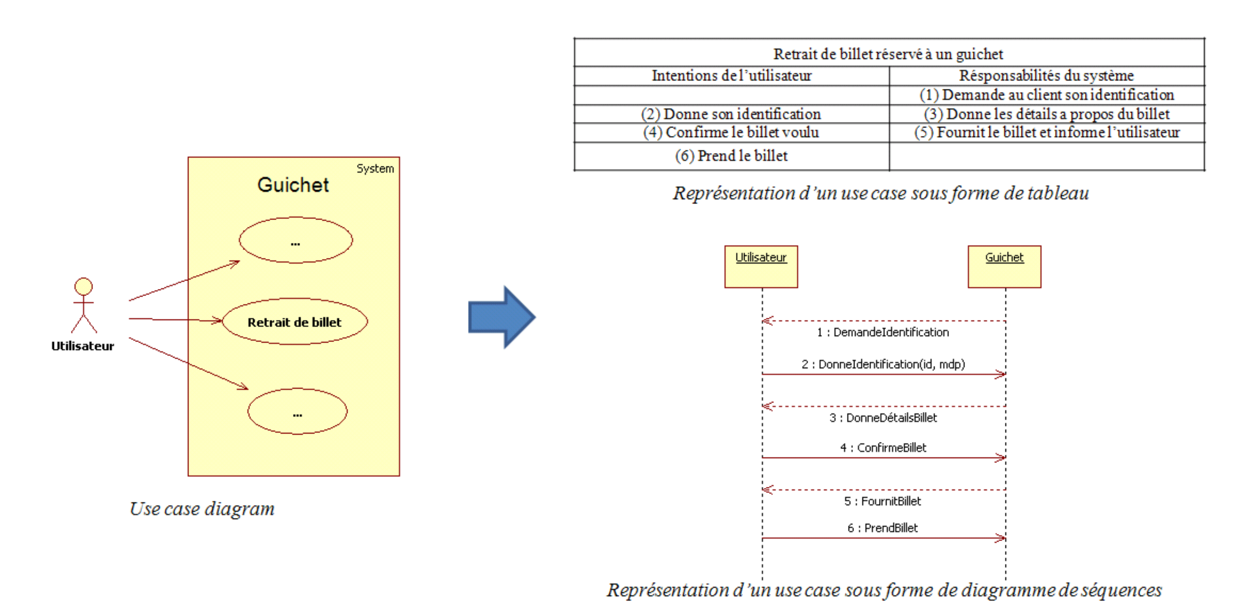
Ce type de modélisation littérale semble cependant plus adaptée pour décrire des buts, des intentions et des tâches relativement abstraites que des tâches d'interaction, ce qui parait long et fastidieux. De plus alors qu'un scénario s'intéresse à un utilisateur en particulier il serait plus intéressant d'abstraire « un scénario type », relatif à un « type d'utilisateur » que l'on aura identifié

A cet effet, Constantine utilise le cas d'utilisation (use case). Dans (Constantine 2001), il en distingue deux types : le cas d'utilisation concret (« concrete use case ») et le cas d'utilisation essentiel (« essential use case », aussi appelé « task case »).

Le « concrete use case » introduit un premier niveau d'abstraction : ainsi, on ne dira plus « Paul écrit son pseudo et son mot de passe dans les champs du formulaire et clique sur soumettre » mais « l'utilisateur renseigne ses identifiants à l'aide d'un formulaire et les soumet ».

« L'essential use case » est encore plus abstrait : il est composé de tâches du type « donne son identification ». Comme on peut le voir dans (Constantine 2001) un « essential use case » est défini comme « représentant des petits morceaux d'exécution d'un rôle utilisateur ». Il se veut essentiel, à savoir simple et abstrait (en contraste avec le scénario qui est complet et réaliste) ainsi qu'indépendant de la technologie (Constantine 2006). On parle alors des intentions de l'utilisateur, et des responsabilités du système. Il est représenté par Constantine sous forme « littérale structurée », à savoir d'un tableau en 2 colonnes (voir Figure 33).

Tout comme nous l'avons vu au travers de l'analyse des activités de conception logicielle, le « use case » peut être représenté graphiquement. Le diagramme de cas d'utilisation permet d'illustrer un ensemble de « use cases » pour un système donné, le diagramme de séquences permettra de décomposer le use case, au même titre que le tableau (figure 17).



| Retrait de billet réservé à un guichet | |
|---|---|
| Intentions de l'utilisateur | Responsabilités du système |
| | (1) Demande au client son identification |
| (2) Donne son identification | (3) Donne les détails a propos du billet |
| (4) Confirme le billet voulu | (5) Fournit le billet et informe l'utilisateur |
| (6) Prend le billet | |



*Figure 33. Différentes représentations graphiques pour la décomposition d'un use case*

Constantine nous met cependant en garde concernant l'utilisation des diagrammes de séquences car elle introduit la considération prématurée de la conception interne du système plutôt

que d'encourager à se concentrer sur la nature essentielle des tâches. Nous retiendrons l'importance de considérer les étapes de la conception une à une en utilisant des formalismes adaptés. C'est ce que fait par exemple (Göransson et al. 2003) en intégrant la « conception de l'utilisabilité » (de l'anglais « Usability Design ») dans les activités du RUP, avant les activités d'analyse et conception. Il relève également les limites du langage UML et plus particulièrement des « use cases » dans cette activité, car ils portent l'attention sur le système et sont de plus « difficiles à comprendre par les utilisateurs ».

D'autres modélisations ont vu le jour, indépendamment des méthodes de conception logicielle mais visant plutôt à outiller celles-ci a posteriori, en caractérisant les tâches IHM : les arbres de tâches.

Les arbres de tâches

La représentation à travers des arbres hiérarchiques permet de décrire tous les niveaux d'abstraction des tâches de la plus abstraite à la plus précise. Nous présentons ici deux modèles de tâches utilisateurs basés sur des arbres hiérarchiques : CTT et K-MAD.

D'après (Mori et al. 2002), « une tâche définit comment l'utilisateur peut atteindre un but dans un domaine spécifique d'application », le but étant « une modification désirée de l'état d'un système ou une requête à ce système ». Les auteurs distinguent quatre types de tâches auxquelles ils attribuent un élément graphique particulier pour créer leur propre formalisme d'arbre de tâches nommé CTT (ConcurTaskTree) et supporté par un éditeur dédié (CTTE)[15] :

- les tâches utilisateur, cognitives ou physiques, entièrement exécutées par l'utilisateur sans interaction avec le système,
- les tâches application, entièrement exécutées par le système (ex. la notification),
- les tâches d'interaction, exécutées par l'utilisateur en interaction avec le système,
- les tâches abstraites, qui ne sont aucune des autres tâches mais sont décomposables en plusieurs d'entre elles.

Les différentes tâches sont liées entre elles de manière à représenter leur chronologie. Ainsi, deux tâches peuvent être indépendantes ou synchronisées, une tâche peut autoriser la suivante ou désactiver la précédente, elle peut être itérative, récursive, voire optionnelle.

(Pribeanu 2005) y ajoute également la notion de couche fonctionnelle (indépendante de la plateforme) et de couche opérationnelle (dépendante de la plateforme). Un exemple d'utilisation de CTTE pour la conception d'interface est illustré Figure 34.

[15] L'outil CTTE et les informations relatives au formalisme CTT sont disponibles à cette adresse: http://giove.isti.cnr.it/tools/CTTE/home

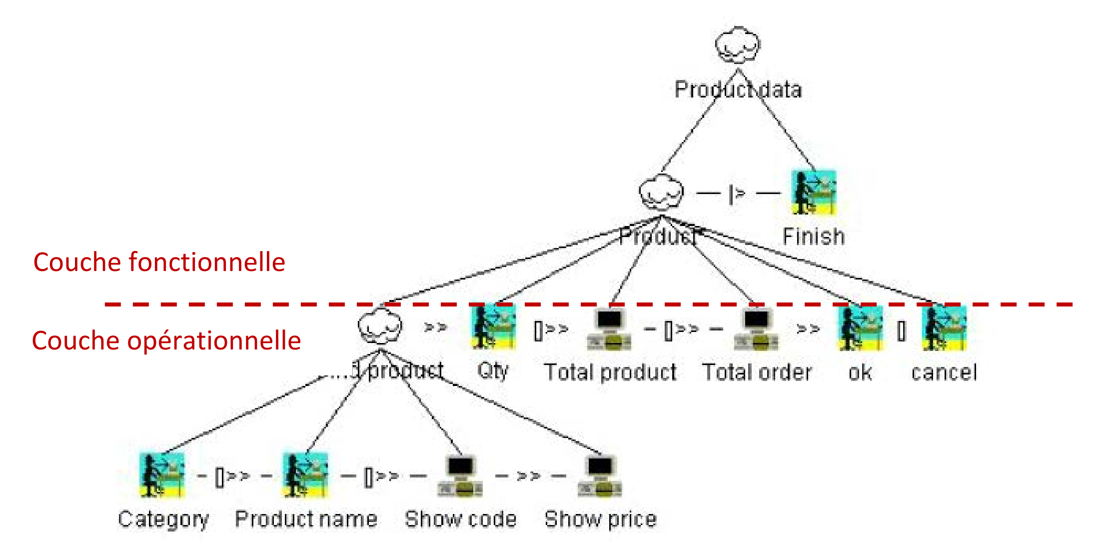



*Figure 34. Exemple d'arbre de tâches CTTE (extrait de dans (Pribeanu 2005))*

Relativement proche de l'éditeur CTTE, l'outil K-MADe[16] (Kernel of Model for Activity Description environnement) (Baron et al. 2006) permet de « représenter l'activité de l'utilisateur sous forme d'arbres de tâches » sur la base d'une structure nommée N-MDA (Noyau du Modèle de Description de l'Activité) (Lucquiaud 2005). Il est composé de trois éditeurs :

- l'éditeur graphique pour construire des arbres de tâches,
- l'éditeur de caractéristiques pour caractériser ces tâches à travers différents attributs,
- l'éditeur des objets manipulés par l'utilisateur.

L'arbre de tâches K-MAD permet de décomposer l'activité en : une tâche générale, des tâches intermédiaires, puis des tâches dites élémentaires (le plus fin degré de granularité). On retrouve le découpage en types de tâches comme dans CTT en fonction de l'exécutant de la tâche :

- la tâche utilisateur, exécutée par un individu ou une organisation,
- la tâche système, exécutée complètement par un équipement,
- la tâche interactive, déclenchée par l'utilisateur et réalisée sur l'équipement,
- la tâche abstraite contient des sous-tâches d'exécutant différent,
- la tâche dont l'exécutant est inconnu est comme son nom l'indique indéterminée.

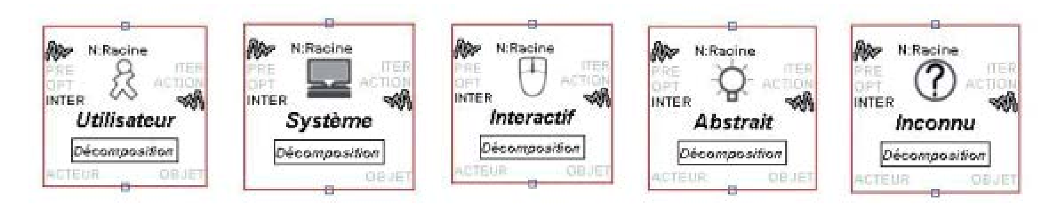



*Figure 35. Les cinq types de tâches qui composent un arbre K-MAD*

Les tâches filles d'une même tâche mère peuvent être ordonnancées de façon séquentielle (une après l'autre), parallèle (en même temps), alternative (une ou l'autre) ou sans ordre. C'est ce qu'on appelle la décomposition d'une tâche (une tâche sans décomposition sera qualifiée

[16] L'outil K-MADe et le manuel utilisateur sont disponibles à cette adresse : http://kmade.sourceforge.net/download.php

d'élémentaire). Une tâche, quelle qu'elle soit, peut être : obligatoire ou facultative, interruptible ou pas, contrainte ou pas (c.à.d.. par un évènement, une condition), itérative ou unique.

Le maquettage

Lors d'un maquettage, il s'agit de représenter graphiquement les fonctions pourvues par le système. Le maquettage représente l'interface à un instant unique.

Un maquettage peut être très précis, avec des éléments graphiques particuliers, un agencement déjà pensé, une gestion de la navigation, etc… Il est un élément important dans les méthodes de conception d'IHM (CCU, CCA, CDB) car il supporte le dialogue entre utilisateurs et concepteurs en leur permettant de représenter la future interface de manière concrète par la composition d'éléments graphiques : les CIO (Concrete Interaction Objects) (ex. Figure 36).

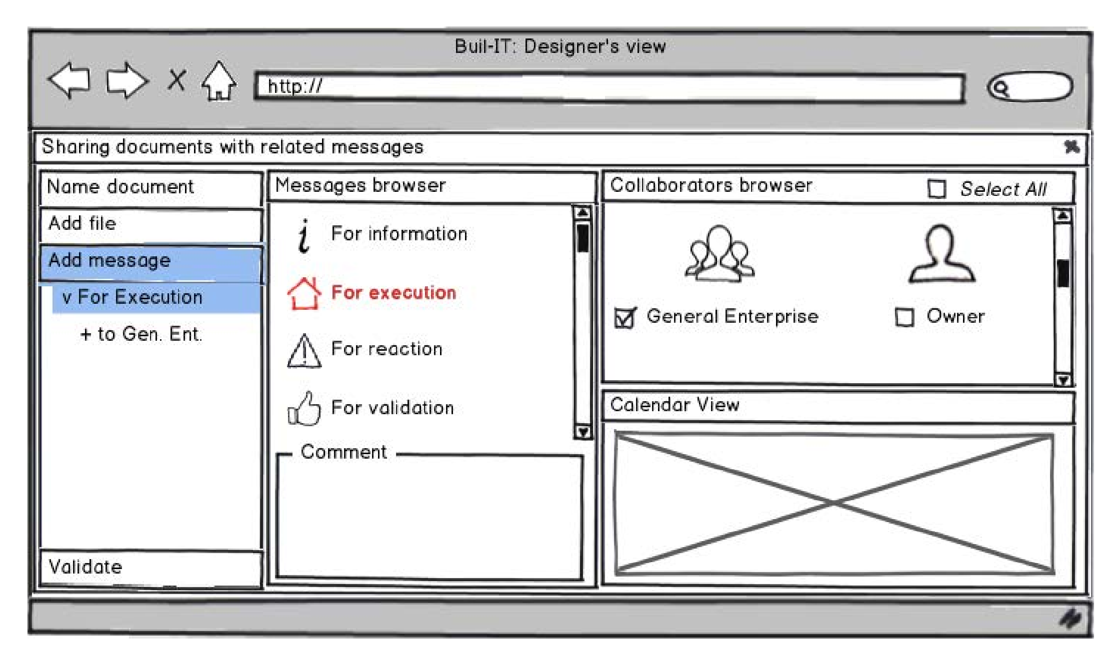



*Figure 36. Exemple de maquettage (outil « Balsamiq Mockup »)*

Ici encore Constantine se démarque par sa volonté d'abstraction en proposant une méthode pour décrire de manière graphique une maquette composée d'objets abstraits (les Abstract Interaction Objects, AIO). Dans (Constantine 2003), il détermine 12 familles de tâches interactives relatives à l'utilisateur ainsi que 11 éléments d'interfaces (des conteneurs et des interacteurs) (Figure 37). En composant ces tâches et des éléments d'interaction sous forme d'éléments graphiques (des « composants abstraits canoniques »), il cherche à modéliser les interfaces à travers des « prototypes abstraits » (Constantine 1998) décrits en termes de forme et de fonction, mais indépendamment de toute apparence (Figure 38). Notons que, dans la composition de plusieurs vues, l'auteur n'illustre pas le lien d'entre les vues. Il s'agit d'illustrer une interface à un instant précis.

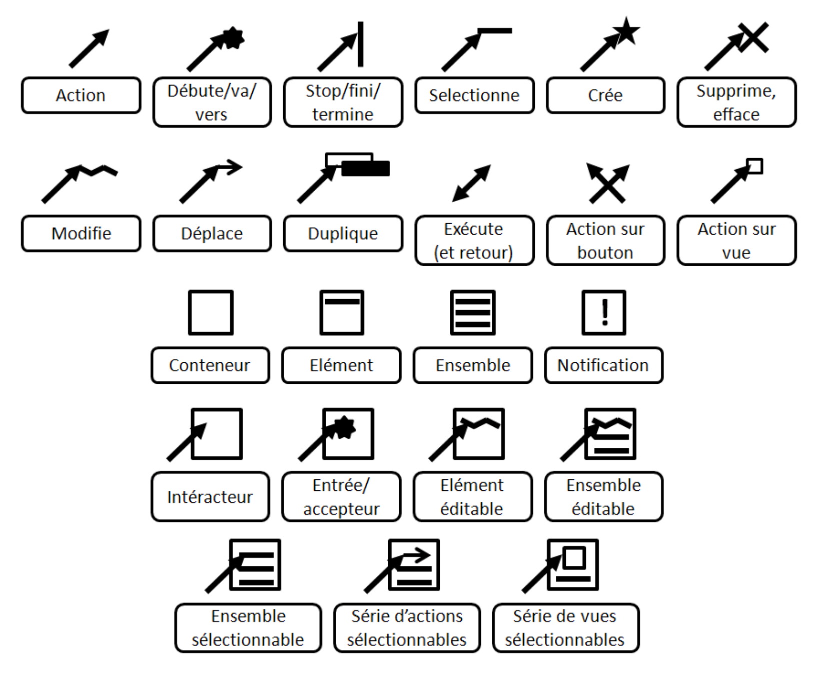



*Figure 37. Tâches et éléments d'interaction abstraits*

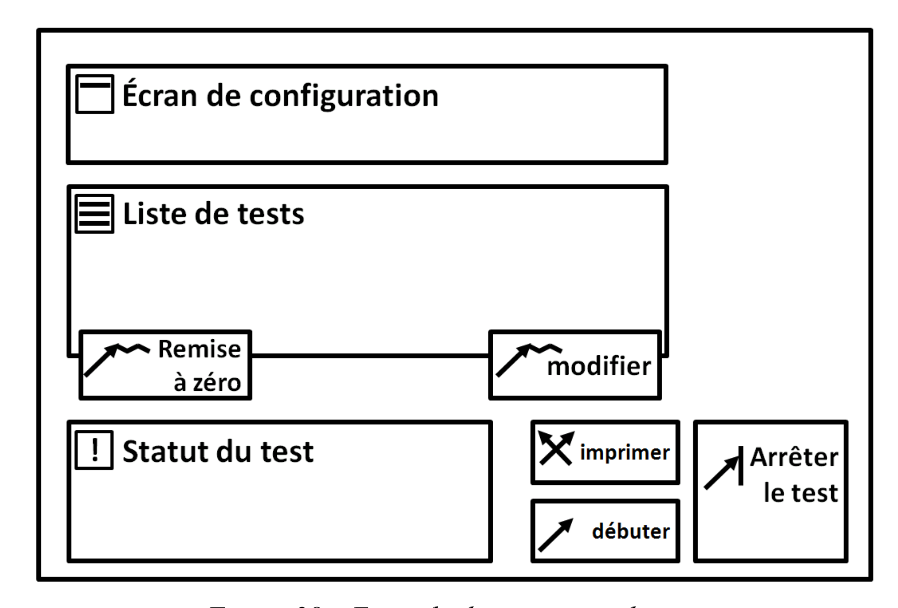



*Figure 38. Exemple de prototype abstrait*

### *1.10.4 Constat et conclusion*

Les méthodes de conception IHM enrichissent la conception logicielle en interrogeant l'utilisateur sur ses préférences et sur ses tâches afin d'habiller les systèmes grâce à des interfaces adaptées. Elles préconisent ainsi l'utilisabilité là où les méthodes de GL traditionnelles se limitent à l'utilité. De manière générale, la méthodologie de la conception logicielle s'appuie sur des modèles qui sont tantôt produits tantôt utilisés par des acteurs différents tout au long des processus. Ces modèles peuvent être concrets (c.à.d. très représentatifs de la future utilisation du logiciel) ou abstraits. Nous constatons d'ailleurs que si les formalismes proposent diverses façons de représenter graphiquement les tâches abstraites comme concrètes, il est rare de trouver une représentation des objets d'interaction de manière abstraite.

Les professionnels du domaine s'accordent à dire que le passage d'un modèle à l'autre reste peu structuré du fait du caractère « manuel » de cette opération. Nous traitons dans la section suivante d'une approche qui, pour pallier à cela, vise la génération, la transformation et l'utilisation de modèles de façon automatisée.

## 1.11 Les enjeux de l'Ingénierie et de l'Architecture Dirigée par les Modèles.

### *1.11.1 Introduction à l'Ingénierie Dirigée par les Modèles (IDM)*

D'un point de vue opérationnel, la conception classique d'un logiciel consiste à comprendre et abstraire son contexte d'utilisation (à la fois professionnel et technique) puis à transférer cette analyse au développeur qui se chargera de la programmation, selon son interprétation. C'est la base des méthodes de conception orientées logiciel et utilisateur présentées dans les sections précédentes, différentes formes de modèles ont été introduites. Cependant, il est possible que la perte d'information ou une mésinterprétation puisse engendrer un manque d'adaptation de la solution par rapport au besoin voire un échec du projet de développement. Le but de l'IDM est de pouvoir automatiser l'implémentation (c.à.d. la génération du code) en réduisant au maximum l'intervention humaine dans l'interprétation de l'analyse et la conception.

« L'IDM se distingue des méthodes de modélisation traditionnelles par la préoccupation constante de rendre les modèles productifs plutôt que contemplatifs » (Favre 2004). Les modèles qualifiés de contemplatifs sont utiles pour la communication et la compréhension mais ne sont pas utilisés pour produire du code, ce qui reste l'activité de l'informaticien. Un modèle devient productif lorsqu'il est rendu interprétable et manipulable par une machine.

Ainsi, l'IDM simplifie le travail des ingénieurs du développement logiciel en leur permettant de ne plus manipuler du code à implémenter mais des modèles de code, à savoir des descriptions écrites dans un langage adapté. On dira que le niveau de modélisation du code est le niveau 0 (ou M0). Il peut être conforme à plusieurs modèles (de niveau M1) eux-mêmes définis par des méta-modèles (le niveau M2) qui s'appuient enfin sur un méta-méta-modèle (le niveau M3). L'IDM englobe plusieurs contextes de travail avec leurs propres concepts et langages, ainsi que les outils et compétences relatives : c'est ce qu'on appelle des « espaces techniques » (ET) (Kurtev et al. 2002; Favre 2004; Pérochon 2008; Bézivin & Kurtev 2005). On distingue par exemple (voir aussi Figure 39) :

- l'ET de la Syntaxe Abstraite : modélisation de programmes exécutables par un langage de programmation (M1) dont la syntaxe est définie dans une grammaire (M2) elle-même conforme à une famille de métalangages (M3) (ex. le langage JAVA, la grammaire JAVA et EBNF (Extended Backus-Naur Form)),
- l'ET XML (eXtendable Markup Langage) : modélisation de la structure et du type de contenu d'un document XML (M1) grâce une syntaxe contrainte par des règles grammaticales et des schémas (M2) conformes à un méta-schéma XML (M3),
- l'ET de l'architecture dirigée par les modèles (MDA pour « Model-Driven Architecture ») : modélisation d'un système d'information et du contexte métier par des modèles et méta-modèles UML conformes au MOF (Meta Object Facility). L'ET EMF (Eclipse Modelling Framework) est similaire à MDA, selon (Jean Bézivin & Kurtev 2005) ils pourraient être considérés comme un seul ET.

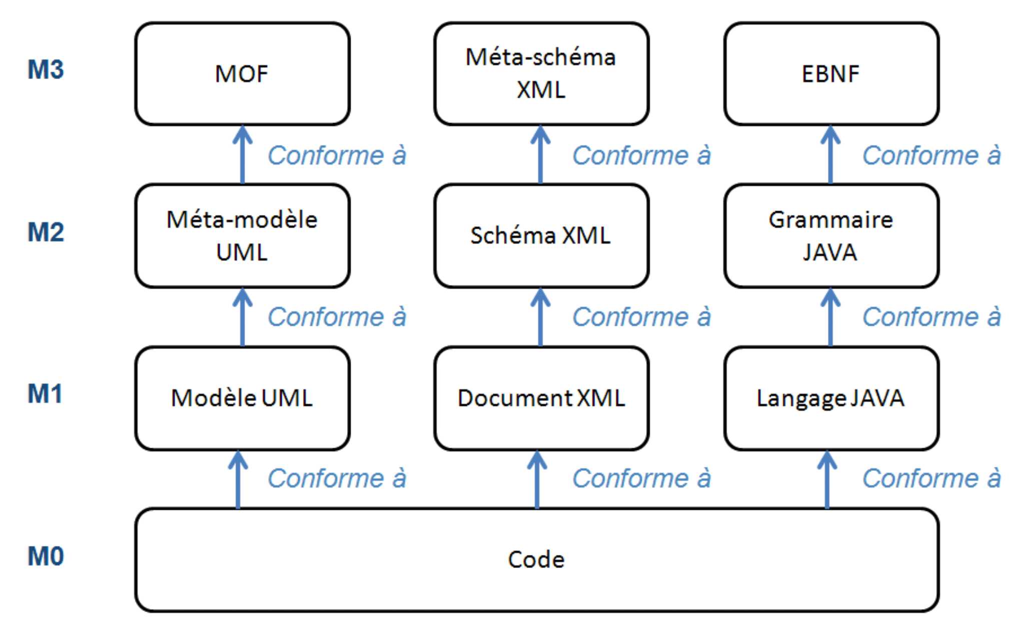



*Figure 39. Espaces techniques de l'IDM et niveaux de modélisation*

Parmi les Espace Techniques de l'IDM, nous nous intéresserons particulièrement à la MDA qui s'applique au développement de systèmes informatiques.

### *1.11.2 L'architecture dirigée par les modèles*

L'architecture dirigée par les modèles[17] (MDA pour « Model-Driven Architecture ») est un « standard industriel » qui consiste à structurer la conception en modélisant le besoin métier indépendamment de toute plateforme technique puis de « transformer » ce modèle en une architecture technique qui pourra évoluer. Cette approche est issue du constat que le métier évolue moins vite que la technique et que l'on doit pouvoir changer de plateforme technique (ou en utiliser plusieurs) pour un contexte métier qui lui ne change pas (ou peu). Le processus de développement se fait à travers quatre modèles UML, chacun étant proposé par un acteur différent (Brown et al. 2005; Favre & Pereira 2007):

- le CIM (Computation Independant Model): c'est le modèle du domaine qui décrit l'environnement d'utilisation du système et fait notamment ressortir les contraintes et les exigences,
- le PIM (Plateform Independant Model) décrit l'architecture du système en représentant le processus d'utilisation et les données manipulées,
- le PSM (Plateform Specific Model) est créé pour implémenter la solution. C'est en fait un ensemble de modèles, le premier étant issu du PIM mais prenant en compte les spécificités de la plateforme, les autres étant le fruit de transformations successives jusqu'à l'obtention de l'ISM,
- L'ISM (Implementation Specific Model) est la spécification du système dans son code source.

Le cadre de référence Caméléon (Calvary et al. 2003) décrit un processus de conception d'interfaces adaptables au contexte d'usage basé sur la MDA. Il associe à chaque modèle un niveau d'abstraction de l'interface, tel que nous l'avons notamment introduit au § 1.10.3. Ces niveaux sont les suivants :

[17] Voir http://www.omg.org/mda

- le niveau tâche-concepts définit les tâches utilisateurs et les concepts manipulés (niveau CIM),
- l'interface abstraite (AUI) définit l'interface en espaces de travail liés entre eux (niveau PIM),
- l'interface concrète (CUI) précise chaque espace en termes de fenêtres et d'objets d'interaction (niveau PSM),
- l'IHM finale (FUI) exécutable (niveau ISM).

(Sottet et al. 2005) illustre un exemple de proposition d'interface basée sur la transformation de modèles. Dans son exemple il cherche à configurer une interface de gestion de température dans plusieurs pièces d'une maison. En suivant les étapes définies plus haut il décrit :

- le CIM par les concepts manipulés comme la température (avec ses valeurs min, max et son unité), les pièces (cellier ou salon) et les tâches à exécuter (choisir, gérer),
- le PIM par le processus d'utilisation à suivre (dérivé des concepts), « choisir pièce » puis « régler température »,
- le PSM par l'interface sous forme de conteneurs et d'intéracteurs liés aux actions du processus. Il s'agit ici d'une fenêtre contenant une liste de sélection de pièces pour « choisir pièce », une liste de sélection de valeurs numériques pour « régler température » et un conteneur affichant l'unité de température utilisée.
- L'ISM, à savoir le code, n'est pas édité : il est généré automatiquement.

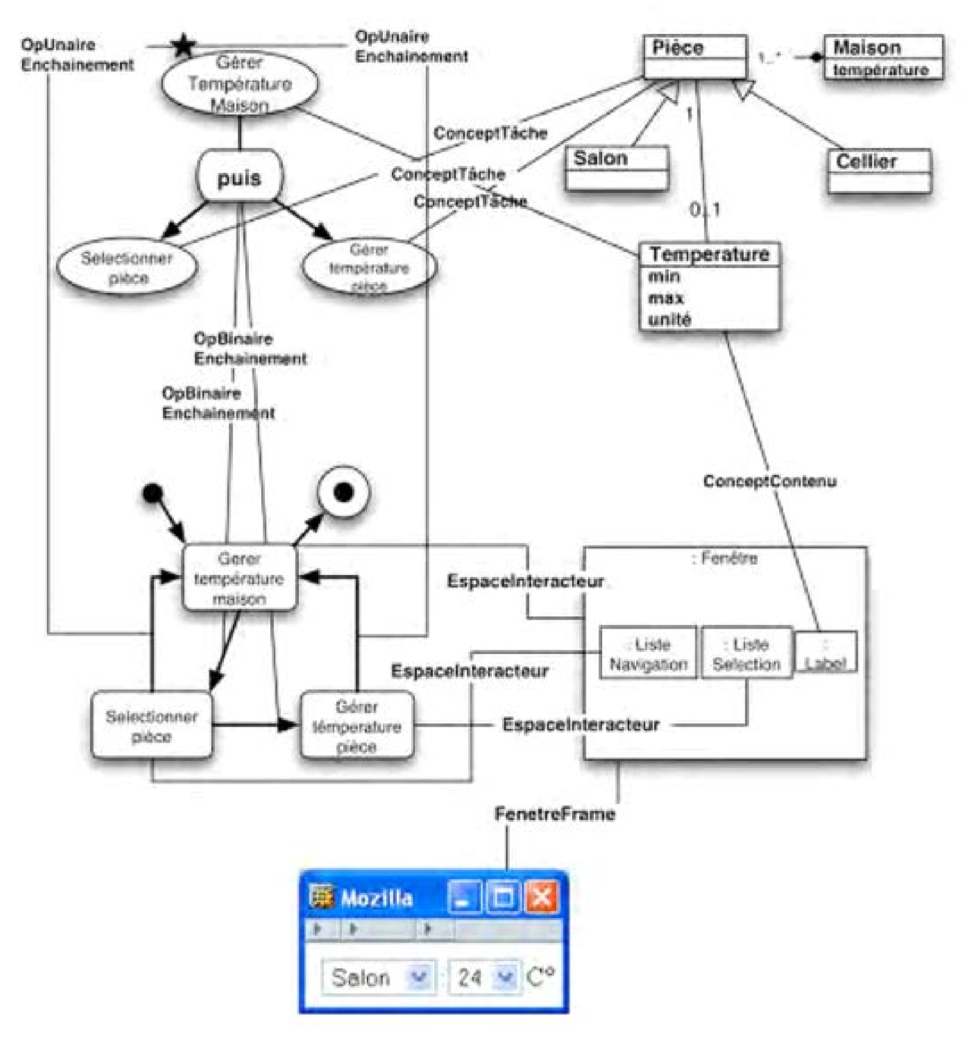



*Figure 40. Illustration des transformations entre modèles dans le cas d'étude de (Sottet et al. 2005)*

L'automatisation des transformations entre les concepts de chaque étape permet de générer rapidement une solution opérationnelle. De plus, tout changement des concepts du CIM induit une potentielle adaptation de la solution générée. Dans l'exemple de (Sottet et al. 2005) que nous venons de décrire, cela pourrait se traduire par l'ajout d'un type de pièce ou encore la modification des valeurs min et max de la température.

Les approches de MDA peuvent prendre différentes formes. Par exemple, (Kalnins et al. 2010) utilise RSL (Requirements Specification Language), un scénario dans « un langage semi-formel pour la spécification des exigences pour un système informatique » dont il déduit le modèle d'analyse sous forme de diagrammes de classes UML et ce, par analyse des mots-clés du scénario. Il constitue ainsi le CIM. Le PIM est créé par l'analyse récursive de tout les cas d'utilisation du scénario et à partir des données du modèle d'analyse. La Figure 41 illustre cette transformation.

Les approches récentes de MDA prennent en compte nombre d'éléments du contexte, relatifs à l'utilisateur, à sa localisation, à son matériel, etc… pour générer des interfaces adaptées au contexte d'utilisation. En ce sens, les premiers modèles ne sont pas réellement indépendants de la technologie. Les contraintes et exigences (modélisées par le CIM) sont en effet liées à l'usage de technologies, ce qui implique que la génération du PIM soit dépendante de cet usage.

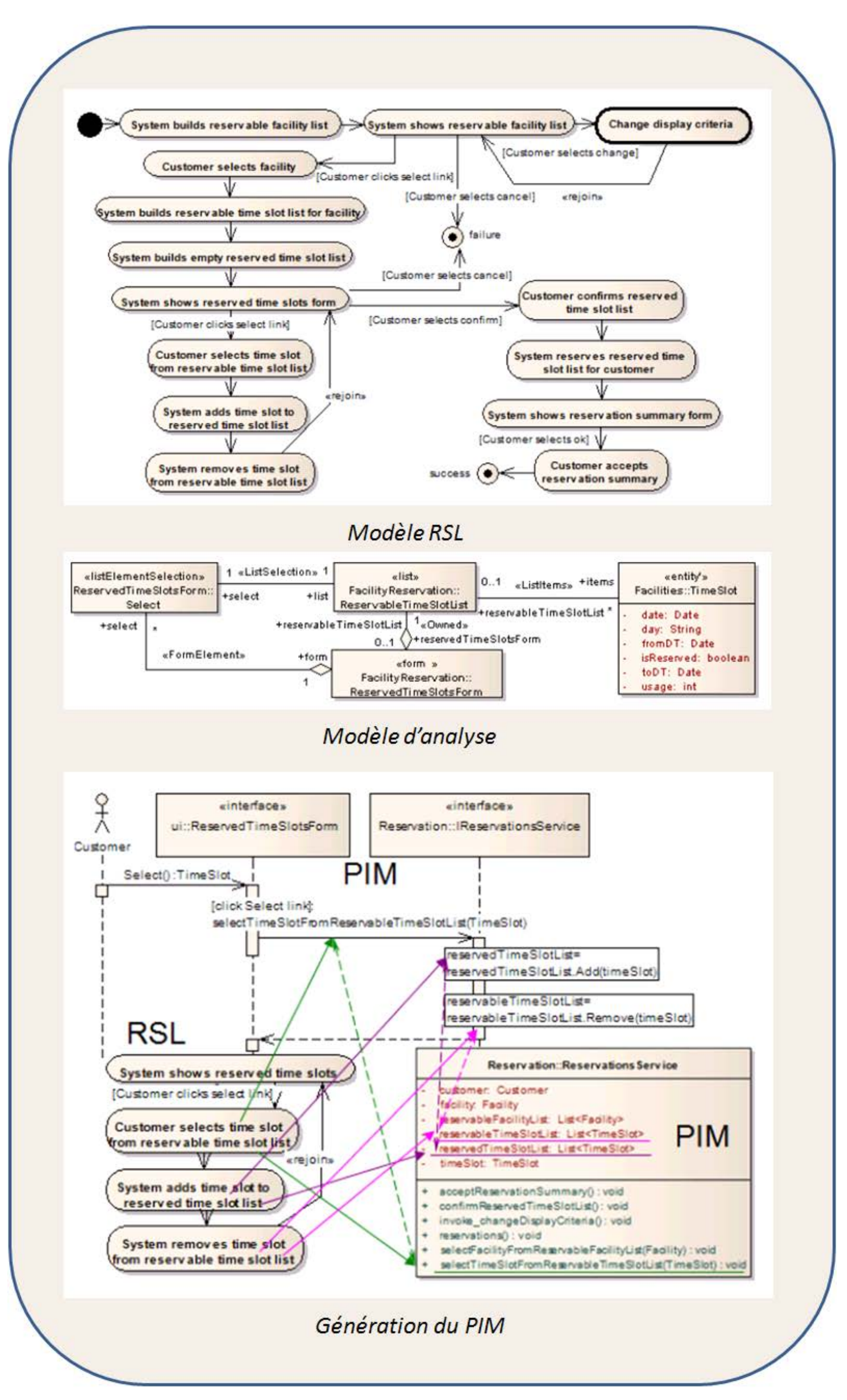



*Figure 41. Les modèles et leur transformation dans l'approche MDA de (Kalnins et al. 2010)*

## 1.12 Conclusion : vers « une méthode centrée usages »

Les théories et méthodes de la conception appliquées au développement logiciel se sont formalisées au cours de la deuxième moitié du siècle dernier.

Aux vues des analyses menées dans ce premier chapitre de la partie 2, le mot clé qui ressort de l'évolution de la conception logicielle est « conceptualisation ». Comme l'illustre le schéma suivant (Figure 42), la première étape fut de conceptualiser le code lui-même à travers des composants réutilisables. Puis le point d'entrée de la conception logicielle s'est déplacé « à des niveaux de plus en plus élevés ». Les concepteurs ont ainsi cherché à d'abord définir les besoins et exigences de l'utilisateur, puis l'utilisateur lui-même au travers de différents modèles. Nous retenons de cette étude l'importance de la **modélisation**. Un concept fort émerge également : il s'agit du concept d'**usage** et de la conception centrée-usages.

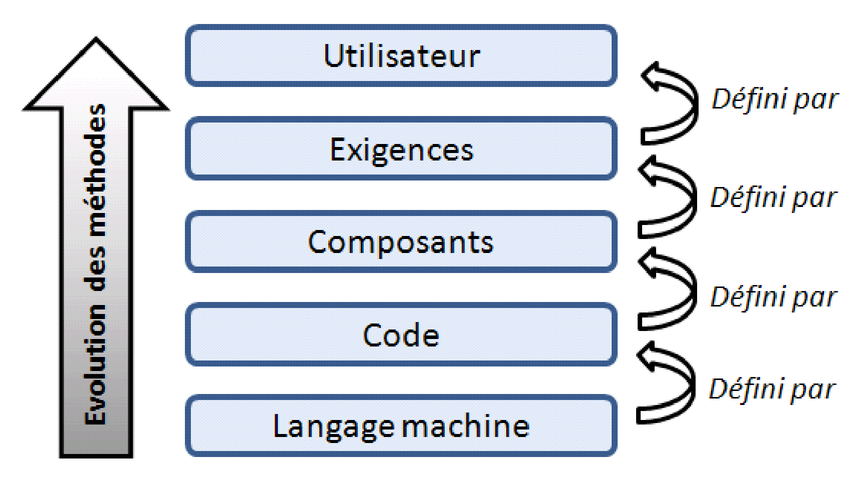



*Figure 42. Points d'entrée de la conception logicielle au cours de l'évolution des méthodes*

Le chapitre suivant montre comment les approches plus récentes « montent d'un niveau » par rapport à l'utilisateur et s'intéressent à son contexte métier. (Dupuy-Chessa 2011) évoque différents paradigmes dont elle résume les apports respectifs en ces termes :

- le GL fournit des techniques et outils (c.à.d. des méthodes) pour mener à bien un projet de conception logicielle,
- le domaine des IHM « nous apprend à considérer l'utilisateur final dès les premières phases d'un projet logiciel »,
- le domaine des SI « prend en compte le contexte organisationnel ».

En effet, les approches de développement dites orientées services ne s'intéressent plus à l'utilisateur « en tant que tel » mais à l'organisation dans laquelle il évolue. Elles permettent alors de répondre aux besoins de l'entreprise en proposant des solutions logicielles décomposées en « services élémentaires pouvant être combinés et réutilisés » (Boucher 2009). Nous nous intéressons notamment à la notion de « processus métier » qui décrit l'orchestration des services dans une entreprise afin d'orienter la conception des systèmes informatiques.

# Chapitre 5 - De l'entreprise orientée services à la conception de Systèmes d'Information

Ce chapitre présente le concept de service dans le monde de l'entreprise puis d'un point de vue informatique. Les services informatiques des technologies de l'information et de la communication (ICT) supportent les processus métiers dans les organisations. Nous nous intéressons à leur conception, de l'analyse de ces processus à la spécification des services.

## 1.13 L'entreprise orientée services

### *1.13.1 Le concept de service*

Un service est une prestation qui met à disposition d'un client « une capacité technique ou intellectuelle » (selon la définition INSEE[18]) pour supporter un besoin. (Maglio & Spohrer 2007) le définit comme « l'application de compétences pour le bénéfice des autres », (Rosemann et al. 2009) précise cette définition en parlant de « capacité autonome et transformationnelle qui est offerte à et consommée par des clients internes ou externes (c.à.d. à l'entreprise) pour leur bénéfice » (voir aussi (Lovelock & Wirtz 1981)). La production d'un service ne fournit donc pas un bien tangible, stockable et consommable a posteriori : elle est immatérielle. Cette production fournit une aide à une ou plusieurs personnes, avec ou sans profit visé (on parlera de services marchands et non marchands), et en fonction d'un contexte métier (ex. les soins médicaux, les transports, la restauration, la vente…). On pourra résumer la définition d'un service comme le fait (O'Sullivan 2006) par trois caractéristiques : c'est une action, qui a une valeur et qui peut composer d'autres services (dits services composites ou méta-services (Crawford et al. 2005)). Au-delà de l'identification des besoins et de la proposition, le cycle de vie de développement d'un service comprend sa vente et sa distribution ainsi qu'un support éventuel (Figure 43).

[18] http://www.insee.fr/fr/methodes/default.asp?page=definitions/services.htm

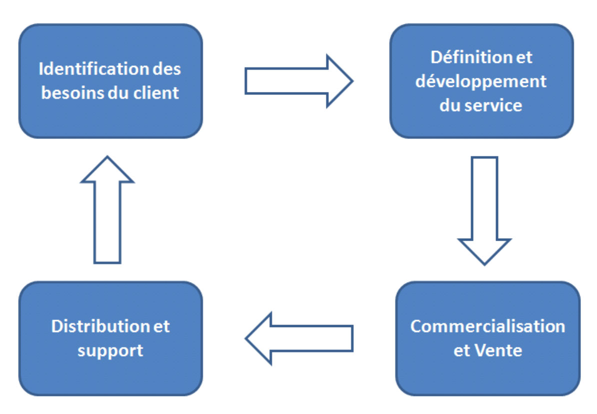



*Figure 43. Le cycle de vie de développement d'un service (tiré de [Birnbaum, 2004])*

Selon (O'Farrell 1991), un service est évalué par : (1) sa **qualité**, définie par la proportion dans laquelle il répond aux attentes du client, et (2) son **utilité** qui est justifiée par l'impact positif qu'aura ce service sur l'activité du client. Notons que ces deux caractéristiques sont indépendantes. En effet, de même que l'exécution parfaite d'un bien n'implique pas de manière évidente qu'on en ait besoin, la qualité d'un service (on parlera de QoS pour « Quality of Service ») n'est pas non plus synonyme d'utilité (ex. un soin prodigué sur une personne saine reste inutile, même si réalisé à la perfection).

Dans la pratique on retrouve la notion service dans plusieurs sciences comme la « business science », « l'information science » ou la « computer science » (Baida et al. 2004).

- Dans la « business science », on parle de service comme nous venons de le définir, à savoir comme d'une « activité métier qui donne lieu à des résultats intangibles et/ou à des bénéfices ». Le terme « e-service » représente un service fourni par le biais d'internet.
- Dans la « computer science », le terme service décrit une fonctionnalité d'une application/d'un logiciel, le web-service est une fonctionnalité accessible via internet.
- Dans « l'information science », on retrouve les termes relatifs aux deux autres sciences : le service dit « informatique » (la fonctionnalité) outille le service dit « métier » (la prestation).

### *1.13.2 L'architecture orientée services*

(Cherbakov et al. 2005) décompose les entreprises en partie opérantes (nommés « business components », c'est-à-dire des « composants métier ») qui regroupent des personnes, des ressources, du savoir-faire et de la technologie et caractérisées par un objectif à atteindre, de par des activités menées, misent en place sous une gouvernance. On parlera de **« services métier »** pour décrire les services échangés entre ces parties opérantes, à savoir tantôt offerts tantôt consommés (en interne et/ou par des clients externes). Une fois orchestrés en **processus**, les services métier sont supportés par un ensemble de « **services ICT** » (Services informatiques de Technologies de l'Information et de la Communication) qui composent le système informatique de l'entreprise : c'est ce qu'on appelle l'**Architecture Orientée Service** (SOA pour Service-Oriented Architecture) (voir Figure 44).

La SOA supporte la proposition de solutions métier et techniques qui pourront évoluer avec le contexte de l'entreprise, en fonction des opportunités de développement ou au contraire des menaces (Zimmermann 2004). (MacKenzie et al. 2006) dira que « dans la SOA, les services sont les moyens par lesquels les besoins et les capacités sont rapprochés ». Autrement dit, elle permet

de mettre les capacités « au service » des besoins. L'évaluation de la QoS contribue à l'évaluation globale de l'entreprise, elle fait partie intégrante de la gestion des entreprises (Cardoso et al. 2004).

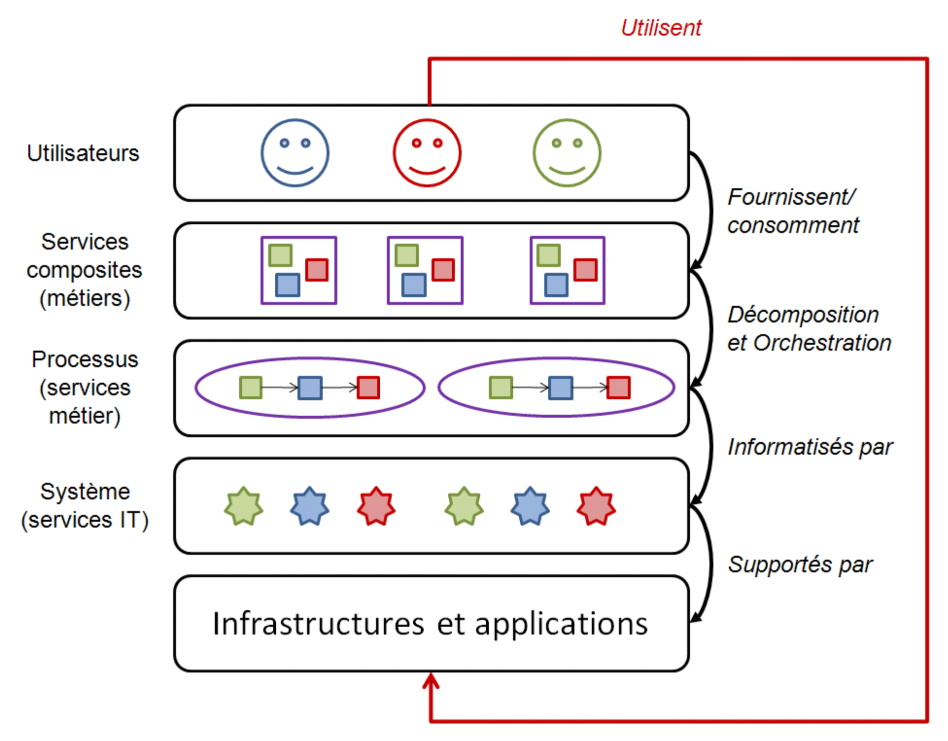



*Figure 44. Services métier et ICT dans la SOA* [19]

Définition d'un service métier

(Lê et al. 2010) propose un langage de description des services métiers (nommé BSDL pour Business Service Description Language) inspiré des travaux de Justin O'Sullivan et son équipe sur la description des capacités et propriétés d'un service (O'Sullivan 2006; Oaks et al. 2003)[20]. Il illustre ce BSDL par un méta-modèle UML (Figure 45) dont les concepts principaux sont les suivants :

- le *fournisseur (ou prestataire)* : entité métier (entreprise, organisation, individu) produisant et fournissant le service,
- le *demandeur (ou client, ou consommateur)* : entité métier (entreprise, organisation, individu) demandant le service,
- la *capacité* : c'est « ce que le service fait ». Les capacités d'un service ont des *règles*, à savoir des *effets* et des *pré-conditions*. Elles possèdent une durée et une date d'exécution. Elles sont décrites lexicalement par des *verbes*.
- La *signature* : c'est un ensemble de *paramètres* représentant les *inputs* et *outputs* d'une capacité (et donc d'un service). Ces paramètres sont décrits lexicalement par des *noms*.
- Les *propriétés* non fonctionnelles : ce sont les *obligations* (telles que le paiement ou le programme) devant être assumées par le fournisseur et le demandeur, engendrant dans le cas contraire une *pénalité*. Le *prix* et la *qualité* sont également des propriétés non fonctionnelles.

[19] http://www.ibm.com/developerworks/webservices/library/ws-WSBFoverviewpart1/index.html
[20] Voir aussi leur site internet : http://www.service-description.com/index.htm

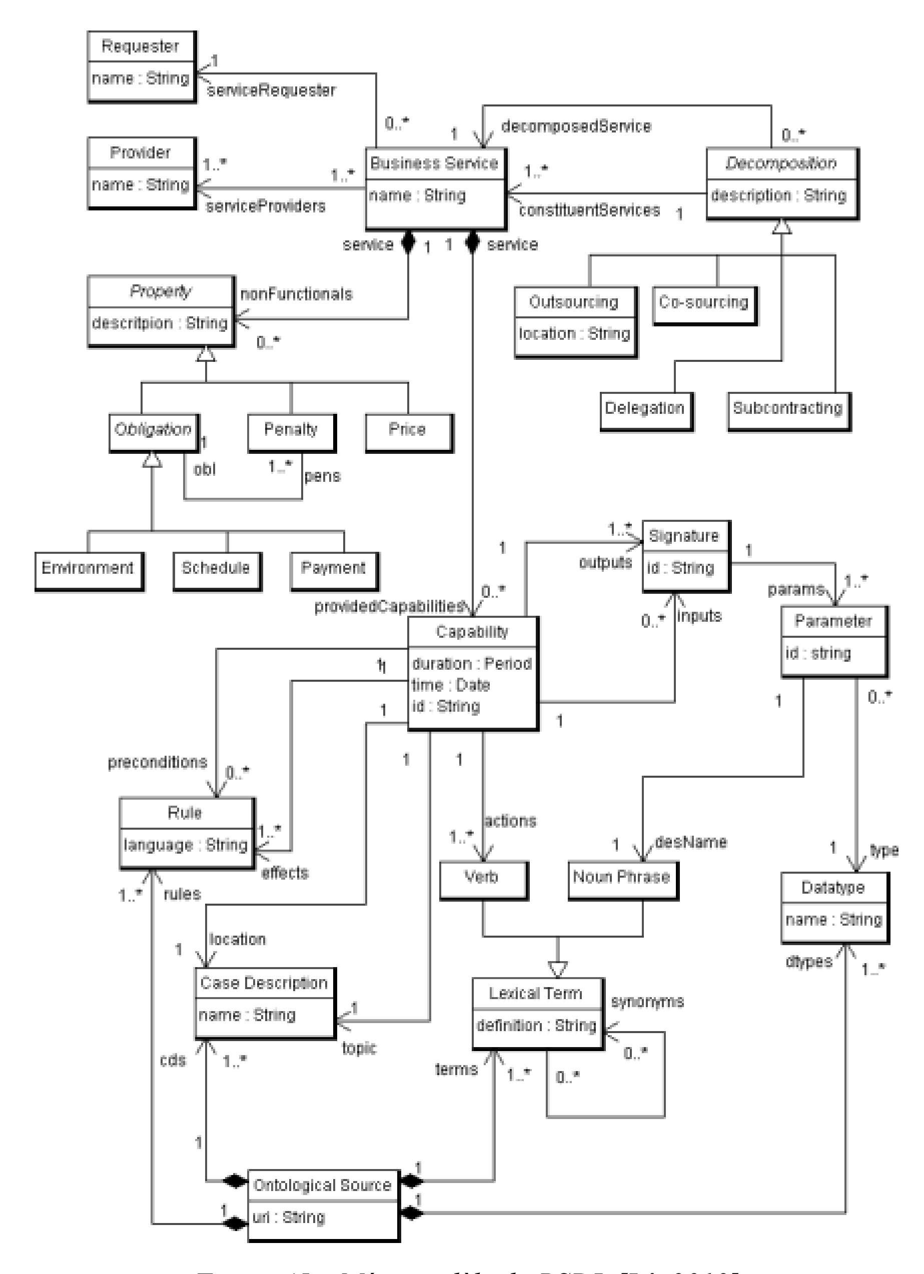



*Figure 45. Méta-modèle du BSDL [Lê, 2010]*

### *1.13.3 Le service informatique ICT*

« En informatique, un service est une fonctionnalité […] mise à disposition par un composant logiciel pour assurer une tâche particulière », il peut être réutilisé ou recomposé en fonction des besoins métiers. Dans une architecture d'entreprise :

- un service ICT « supporte l'exécution d'un service métier » (Kohlborn et al. 2009),
- un service ICT est délivré par un système[21] informatique pouvant être utilisé séparément par plusieurs entités.

[21] (Wieringa 1998) définit un système comme un ensemble d'éléments interactifs, incluant le matériel (hardware) et le logiciel (software). Le système informatique n'est autre que le support technologique du système d'information d'une entreprise.

Un service informatique, au même titre qu'un service métier est défini par : un client, un fournisseur et une description (comprenant Capacités, Paramètres, et Propriétés non fonctionnelles). La description n'est plus lexicale mais adopte un langage informatique compris par les systèmes. La distribution d'un service informatique se fait telle qu'illustrée dans la figure 46, suivant le paradigme « trouver, joindre, invoquer » (Endrei et al. 2004; Papazoglou & Georgakopoulos 2003) : le fournisseur publie le service dans un registre qui permet au client de le trouver et ainsi d'y accéder. Le client y accède grâce à un programme, installé sur un dispositif et proposant une interface logicielle adaptée (Figure 46).

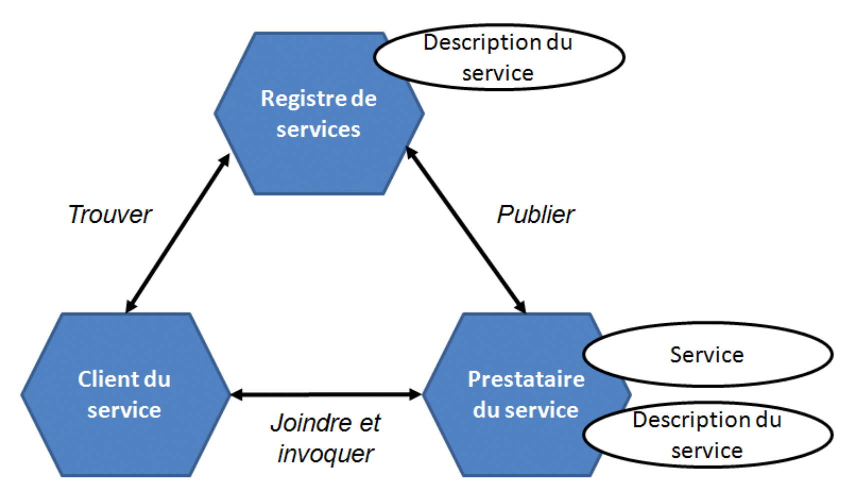



*Figure 46. Rôles et interactions dans une SOA (tiré de (Endrei2004))*

L'approche de développement basée sur les services a été guidée par la volonté de structurer, développer et déployer les logiciels, de manière plus flexible tout en réduisant les coûts et la durée. En 2000, (Bennett & Layzell 2000) faisait émerger plusieurs enjeux relatifs à un développement logiciel « souple » : la réponse aux exigences de l'utilisateur, la personnalisation, l'adaptation, une structure simple, la transparence ou encore la confiance. Ils introduisaient alors une approche permettant aux consommateurs de « choisir la combinaison de services la plus appropriée à tout moment » (on parlera du concept de « late binding »).

Dans cette idée, (Sommerville 2005) redéfinit le concept d'ingénierie des exigences (introduit en 4.1.2) comme la définition du système en termes de services fournis et utilisés. Il encourage ainsi la construction par la configuration de services existants, permettant d'augmenter la rapidité de la formulation des exigences mais aussi de leur interprétation et du développement.

Tout comme d'un point de vue organisationnel la SOA encourage le développement de l'entreprise par sa réorganisation, elle permet d'un point de vue Ingénierie Logicielle de réorganiser et redéployer des processus métiers, des composants fonctionnels et des informations sous forme de services autonomes (Van Den Heuvel 2009; He 2003). En effet, la réduction de l'interdépendance entre les services (le « loose-coupling ») permet de réduire les risques induits par le changement d'un élément sur le reste du système. Ainsi, l'évolution partielle du contexte organisationnel d'une entreprise n'aura d'impact que sur les services concernés et non sur l'intégralité du système.

Le cas particulier des services-web.

Avec la généralisation de l'utilisation des réseaux et plus largement d'internet, autant dans le milieu professionnel que dans les foyers, l'accès aux services métiers par le biais des services ICT est désormais commun. Le client d'un service peut aujourd'hui, où qu'il soit, être « intégré » dans l'architecture d'entreprise et « trouver » son service, en tant qu'utilisateur de services informatiques connectés via le web : on parle alors de « services-web ». Le service-web peut être défini littéralement comme « une interface qui décrit une collection d'opérations qui sont accessibles par un réseau » (Gottschalk et al. 2002). Il est décrit grâce à un langage nommé WSDL (Web Service Description Language) et stocké dans les serveurs du fournisseur puis publié dans un registre UDDI (Figure 47) qui pointera le bon service en fonction de la demande du client. Le client accède alors au service via un protocole SOAP/http. On distingue 3 catégories de services-web[22] :

- les services-web d'intégration donnent accès aux services de transactions : le client accède directement aux services de vente (ex. amazon, itunes…), de gestion de ses comptes bancaires, d'organisation de ses voyages (ex. booking, lastminute…),
- les services-web d'information étendent la portée des structures fournissant de l'information (actualités, météo, finances…),
- les services-web d'externalisation permettent à plusieurs structures d'associer leurs processus en les « faisant sortir » de leur propre architecture (pour par exemple combiner la manufacture, l'assemblage, la distribution…).

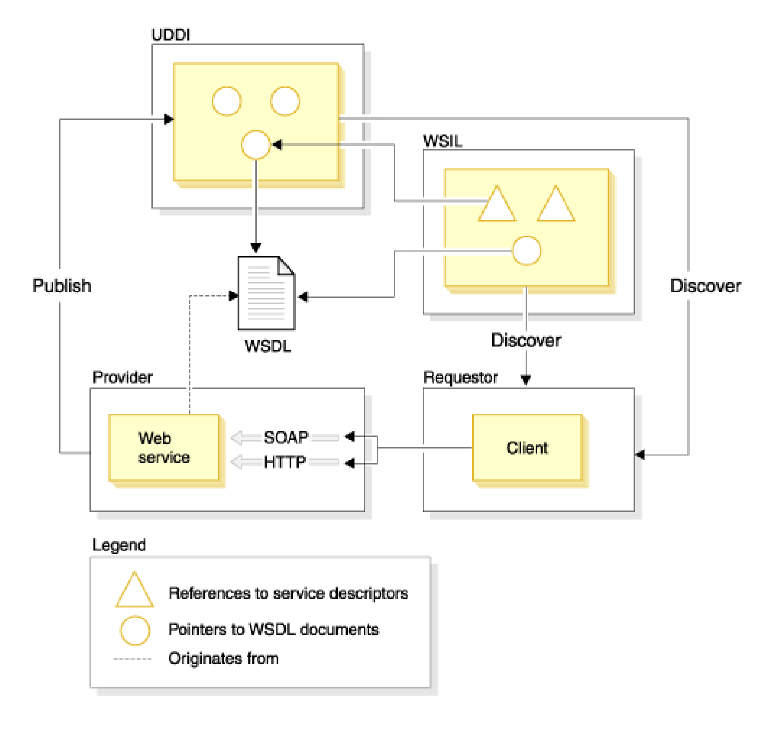



*Figure 47. Rôles et interactions dans une architecture de services web[22]*

22 http://publib.boulder.ibm.com/infocenter/rtnlhelp/v6r0m0/index.jsp?topic=%2Fcom.ibm.etools.webservice.doc%2Fconcepts%2Fcws.html

### *1.13.4 La place des services ICT dans un business model compétitif*

Rappelons qu'une entreprise est définie comme une structure qui combine en interne des moyens humains, matériels, immatériels (services) et financiers[23], afin de créer et fournir en externe des biens ou d'autres services. On distinguera alors entreprises de manufacture et entreprises administratives, tout comme on pourra parler d'industrie traditionnelle et d'industrie de services. Mais quelle que soit leur nature, elles adoptent toutes un « Business Model » particulier.

Afin de rester compétitives, les entreprises doivent maintenir voire augmenter leur productivité (c.à.d. l'efficacité, le rendement de leur processus de création de biens ou de services) et leur production (c.à.d. l'apport de valeur ajoutée à cette création). Pour cela, la restructuration de leur modèle (le « Business Model) est parfois nécessaire. D'un point de vue organisationnel, cela implique l'adoption de nouveaux produits et processus (Hodgson 2003). Les impacts respectifs du Taylorisme (fin du 19e siècle) et du Fordisme (début du 20e) sur la croissance du secteur de la sidérurgie et de l'automobile en sont un exemple connu. Les modèles furent en effet bouleversés de par l'évolution des processus de travail et des produits avec l'application de plusieurs principes tels que[24] :

- la division du travail à la fois verticale (c.à.d. la séparation de la conception et de la réalisation) et horizontale (c.à.d. la répartition des tâches, avec l'apparition du travail à la chaîne),
- ou la standardisation des pièces favorisant la production en série.

Mais actuellement, l'optimisation des processus ne suffit plus à la croissance dans un environnement toujours plus demandeur doublé d'un contexte économique concurrentiel et changeant : il faut vendre plus, moins cher, en dépensant moins… Dans son étude sur « la place du business model dans la compagnie », (Osterwalder 2004) montre que l'évolution de ce dernier est dépendante, non seulement de l'organisation (c.à.d. des processus métier, comme décrit plus haut) mais aussi de la stratégie adoptée (définie par une position, des objectifs…) et des Systèmes d'Information utilisés. La Figure 48 illustre cette corrélation, ainsi que l'influence de ce que l'auteur appelle « forces extérieures », relatives à l'environnement (compétition, changements sociaux, légaux, technologiques ou dans la demande des clients). D'un point de vue stratégique, la compétitivité est assurée par la conquête de nouveaux marchés et par l'adoption de nouvelles compétences. D'un point de vue technologique, c'est l'adoption de nouvelles solutions en termes de services ICT qui aura un impact positif. (Osterwalder 2004) le justifie à travers les quatre apports suivants :

- la réduction des coûts de transaction et de coordination,
- l'offre de nouveaux produits et services plus complexes de par la collaboration de plusieurs compagnies,
- l'amélioration et l'expansion des moyens d'atteindre le client,
- la proposition de nouveaux ordres de prix et de mécanismes de revenus.

[23] http://fr.wikipedia.org/wiki/Entreprise
[24] http://fr.wikipedia.org/wiki/Fordisme

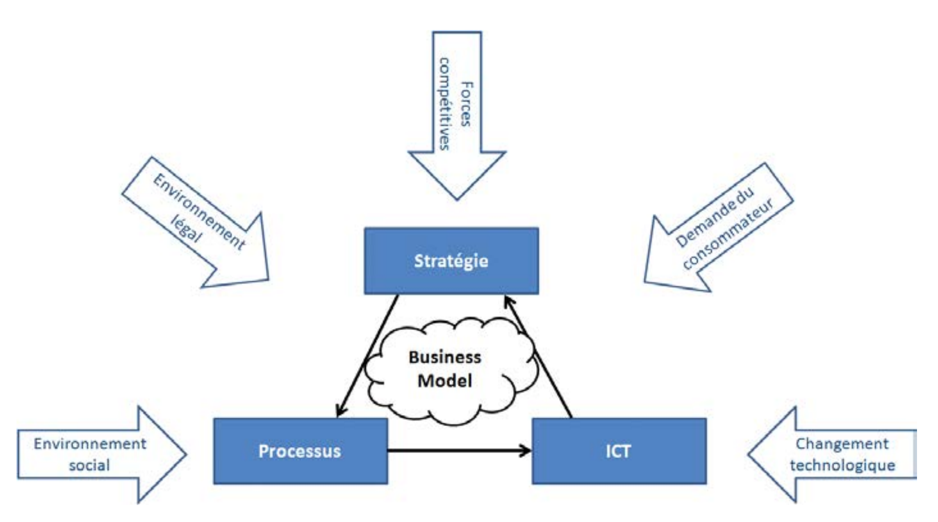



*Figure 48. Corrélation entre Business model, Environnement, Stratégie, Processus et Systèmes d'Information [Ostwalder, 2004 - p.16]*

La valeur des services ICT n'est donc plus à prouver dans un business model qui se veut compétitif. Dans la section suivante, nous nous intéressons particulièrement à leur conception à partir de la modélisation des processus métiers.

## 1.14 De la modélisation des Processus Métier au Système d'Information

### *1.14.1 Introduction à la conception de Systèmes d'Informations (SI)*

Un système d'information est la combinaison des activités des membres d'une entreprise (c.à.d. qui supportent les opérations métiers, la gestion et la prise de décision) et des technologies de l'information qu'ils utilisent (c.à.d. pour stocker, récupérer et transmettre de l'information).

(Zachman 1987) s'est inspiré du domaine de la construction pour conceptualiser l'architecture des systèmes d'information. Son framework permet de décrire un SI en adoptant plusieurs points de vue :

- le point de vue du planificateur dans laquelle sont définis les objectifs,
- le point de vue du maître d'ouvrage qui décrit le contexte métier de façon conceptuelle,
- le point de vue du concepteur qui décrit le système d'information de façon précise,
- le point de vue de l'ingénieur qui décrit les contraintes techniques,
- le point de vue du constructeur qui décrit les consignes de développement,
- le point de vue machine, à savoir le code implémenté.

Zachman introduit également une série de questions à se poser pour décrire SI. Ce questionnement est mené pour chaque point de vue et donne lieu à un ensemble de modèles descriptifs pour chacun d'eux. Il obtient ainsi un schéma de classification en 2 dimensions pour la représentation descriptive d'une entreprise (figure 25). Les éléments qu'il cherche à déterminer et leurs questions relatives sont :

- le « Quoi ? » qui définit les données,
- le « Comment ? » qui interroge les processus fonctionnels,
- le « Où ? » et le « Quand ? » qui déterminent le contexte de localisation et temporel,
- le « Qui ? » qui définit les acteurs et leurs responsabilités dans l'organisation,
- et le « Pourquoi ? » qui a pour but d'identifier le but, les motivations.

Comme l'illustre (Frankel et al. 2003) (voir Figure 49), les méthodes de génie logiciel et notamment l'architecture dirigée par les modèles (MDA) peuvent être intégrées à cette approche, le passage d'un point de vue à l'autre permettant de guider le processus de conception des technologies de l'information (CM Pereira 2004).

- Le CIM (Computation Independant Model) est le point de vue de l'Analyste Métier qui modélise le métier
- Le PIM est le point de vue du Concepteur qui modélise le système
- Le PSM et l'ISM (le code) sont le point de vue du Développeur qui modélise la solution technologique et la développe.

Abstractions (Columns)

Perspectives (Rows)

| The Zachman Framework | DATA *What* *(Things)* | FUNCTION *How* *(Process)* | NETWORK *Where* *(Location)* | PEOPLE *Who* *(People)* | TIME *When* *(Time)* | MOTIVATION *Why* *(Motivation)* |
|---|---|---|---|---|---|---|
| SCOPE (Contextual) *Planner* | List of things important to the business | List of processes the business performs | List of Locations in which the business operates | List of Organizations Important to the Business | List of Events Significant to the Business | List of Business Goals/Strategies |
| BUSINESS MODEL (Conceptual) *Owner* | Semantic Model | Business Process Model | Business Logistics System | Work Flow Model | Master Schedule | Business Plan |
| | Computation-Independent Model (CIM) | | | | | |
| SYSTEM MODEL (Logical) *Designer* | Logical Data Model | Application Architecture | Distributed System Architecture | Human Interface Architecture | Processing Structure | Business Rule Model |
| | Platform-Independent Model (PIM) | | | | | |
| TECHNOLOGY MODEL (Physical) *Builder* | Physical Data Model | System Design | Technology Architecture | Presentation Architecture | Control Structure | Rule Design |
| | Platform-Specific Model (PSM) | | | | | |
| DETAILED REPRESENTATIONS (Out-of-Context) *Sub-Contractor* | Data Definition | Program | Network Architecture | Security Architecture | Timing Definition | Rule Specification |
| | CODE | | | | | |

*Figure 49. Le Framework Zachman et les modèles de la MDA (extrait de (Frankel 2003)*

<u>Un nouveau paradigme</u>

Comme on peut le lire dans (Ramollari & Dranidis 2007), le développement orienté service est un changement de paradigme dans l'ingénierie logicielle dans lequel « le service est une abstraction utilisée pour supporter le développement rapide et peu onéreux d'applications à travers la composition de services ». Nous comprenons ici que la conception d'un SI est une activité de conception logicielle, qui suit les principes de développement utilisant des techniques d'abstraction par les modèles (Loniewski et al. 2011), mais qu'elle adopte un autre point de vue.

Selon (Papazoglou & Van Den Heuvel 2006), la méthodologie de la conception et du développement orientée services (SODD) se base sur les modèles de développement logiciel tels que le développement orienté composant et le RUP (voir 2.1.1) tout en se concentrant sur la modélisation des processus métiers dans le domaine de l'entreprise. Dans le même ordre d'idée, (Hussain et al. 2010) effectue également un mapping entre le framework de développement de service SOMA (Service-Oriented Modeling and Architecture) (Arsanjani 2004) et le RUP (cf. § 1.9) en mettant en parallèle les phases de conception de service du framework (Identification,

Spécification, Réalisation, Déploiement) et les phases de conception logicielle du RUP (Inception, Elaboration, Construction, Transition).

Trois niveaux d'abstraction de la conception de SI sont présentés dans (Front et al. 2009) :
- le niveau intentionnel qui décrit les besoins et les buts pour en déduire les exigences du système,
- le niveau organisationnel qui décrit « l'organisation à mettre en place pour répondre aux objectifs »,
- le niveau opérationnel qui décrit les solutions informatiques.

On retrouve au travers de ces trois niveaux non pas les phases du RUP mais plutôt la caractérisation des différentes activités. Aux niveaux intentionnel et opérationnel, se déroulent les activités de modélisation du domaine, d'ingénierie des exigences et de développement au travers des différents modèles présentés précédemment.

Le changement de paradigme apparait au niveau organisationnel, au cours des activités d'analyse et de conception : là où les domaines du GL et des IHM définissent des « cas d'utilisation » et des « tâches » relatives à un acteur, l'ingénierie des SI requiert la modélisation des activités de chacun des acteurs impliqués dans des « processus » relatifs à une organisation.

La section suivante présente les formalismes utilisés pour la modélisation d'une organisation et des processus. Nous nous intéressons ensuite au déroulement de cette phase d'analyse et conception.

### *1.14.2 La modélisation du métier et des processus*

Nous avons déjà parcouru un ensemble de modèles utilisés pour représenter l'architecture d'un système dans les méthodes de GL et IHM. Nous nous intéresserons ici plus particulièrement à la modélisation des processus métiers qui est la clé des méthodes orientées services.

« Les processus métiers sont considérés comme des blocks réutilisables indépendants de l'application et de la plateforme qui la supporte » (Papazoglou & Van Den Heuvel 2006). Un processus peut être décrit à travers quatre perspectives (Curtis 1992) :
- la perspective fonctionnelle qui décrit les règles métier,
- la perspective comportementale qui décrit le séquençage des activités,
- la perspective organisationnelle qui décrit les acteurs,
- la perspective informationnelle qui décrit les éléments d'information.

(List 2006) ou encore (Aguilar-Savén 2004) décrivent et évaluent plusieurs langages de modélisation des processus métiers (BPMLs). Les plus largement répandus sont : les diagrammes d'activités UML, BPMN et IDEF3.

Les diagrammes d'activité UML (AD UML) et les diagrammes BPMN (Business Process Modelling notation) sont souvent comparés et confrontés dans la littérature (White 2004; Eloranta & Kallio 2006; Peixoto et al. 2008). BPMN et AD sont deux vues (ou instances) du même méta-model, le BPDM (Business Process Definition Metamodel). Ces diagrammes représentent dans des « swimlanes » les activités de différents acteurs sous forme d'évènements, d'états et de données manipulées. Ces deux formalismes sont très similaires, malgré que le diagramme

d'activités UML reste plus technique et donc moins accessible aux business analystes. En effet, si BPMN est un langage qui fut développé exclusivement pour la représentation des processus métier, les diagrammes d'activité UML décrivent originellement les activités et actions d'un système. C'est pourquoi il aura fallu étendre ce formalisme en stéréotypant les activités en processus (il s'agit de créer un élément nouveau, dérivé de celui qui existe déjà, mais qui a des propriétés spécifiques adaptées à un usage spécialisé, ici la représentation des processus métier). BPMN et AD suivent tout deux les patterns des workflow de (Huang & Mynatt 2006), que l'on peut regrouper en quatre familles : les contrôles de flux, les données, les ressources, les exceptions.

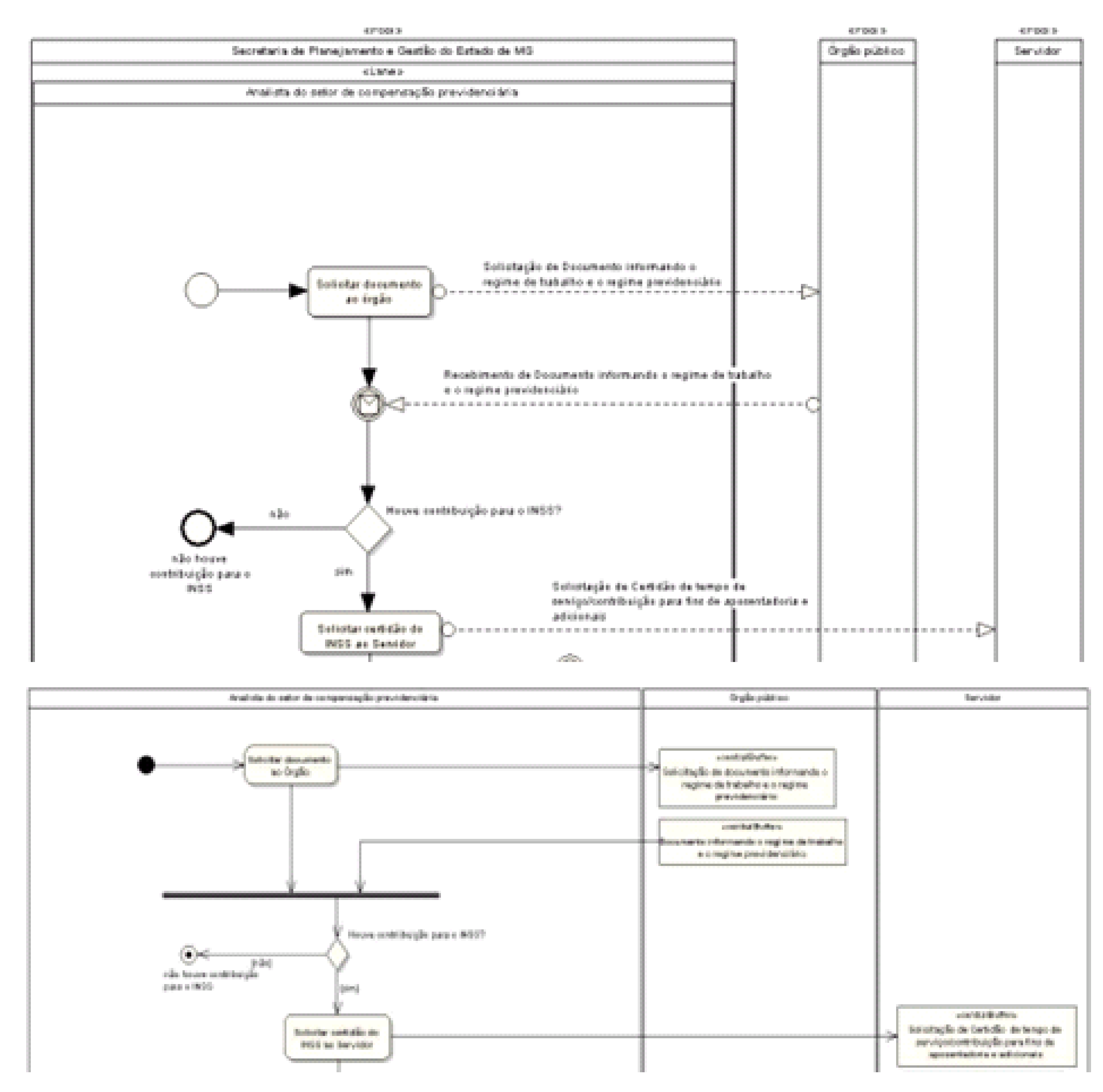



*Figure 50. Représentations d'un même processus avec BPMN (en haut) et AD UML (en bas)*

Comme cité précédemment, le langage UML doit être étendu pour correspondre aux besoins de représentations du métier. De manière générale, sept diagrammes UML peuvent être étendus et utilisés pour représenter des éléments d'un business model (Cesare & Lycett 2003) (Tableau 8).

Tableau 8. Les diagrammes UML et leur potentiel de modélisation du métier (Cesare & Lycett 2003)

| | | Potentiel de modélisation du métier | | | | | |
|---|---|---|---|---|---|---|---|
| Diagramme… | Descriptions | Rôles | Acteurs | Activités | Intéractions | Buts | Entités |
| …de cas d'utilisations | Modélise la fonctionnalité d'un système tel qu'il est perçu par les acteurs externes qui appellent cette fonctionnalité | ■ | ■ | | | ■ | |
| …de classes | Modélise la partie statique d'un système en termes de classes, relations entre classes et d'aggrégation de celles-ci | ■ | | | | | ■ |
| …d'objets | Modélise la représentation des instances des modèles de classes | ■ | | | | | ■ |
| …de collaboration | Modélise les objets d'intéraction à travers des transfert de messages, mettant l'accent sur les roles dans l'interaction et les liens entre chaques | | ■ | | ■ | ■ | ■ |
| …de séquence | Modélise les objets d'intéraction à travers des transfert de messages, mettant l'accent sur les séquences d'intéractions | | ■ | | ■ | ■ | ■ |
| …d'étât | Modélise les états et les transitions d'un objet d'une classe donnée | | | ■ | | | ■ |
| …d'activité | Modélise les flux d'activité pour une procédure donnée | | | ■ | | ■ | |

(Noran 2000) confronte UML et IDEF (ICAM[25] DEFinition) qui est aussi une suite de méthodes de modélisation et supporte la conception de systèmes et l'analyse métier en 3 étapes :

- IDEF3 (pour la description de processus et des objets) et IDEF5 (pour la description d'ontologies) modélisent « le monde réel » et les relations entre les acteurs, les évènements…
- IDEF0 (modélisation des fonctions) et IDEF1 (modélisation de l'information) modélisent les exigences relatives à la gestion de l'information
- IDEF1x (modélisation des données), IDEF2 (modélisation de la dynamique des systèmes) et IDEF4 (Conception Orientée Objet) supportent la conception du système répondant aux besoins identifiés précédemment.

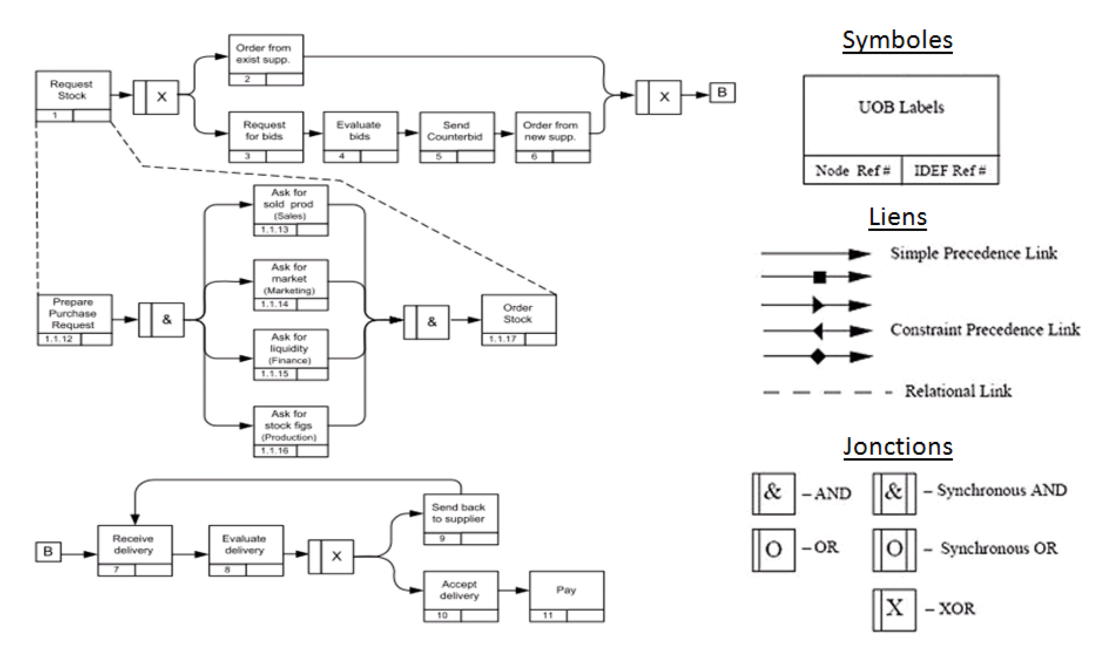



*Figure 51. Un exemple de diagramme IDEF3 pour la description des processus (Noran 2000)*

[25] Pour Integrated Computer Aided Manufacturing, un programme de l'US Air Force

### *1.14.3 Du service métier au service informatique*

Comme nous l'avons dit, les activités du niveau organisationnel supportées par les processus dans le développement d'une SOA, correspondent aux activités d'analyse et conception au cours d'un développement logiciel. Rappelons que dans le RUP, l'analyse consiste à modéliser le « quoi faire » puis lors de la conception on modélise le « comment faire ». Dans le développement d'une SOA , on retrouve cette distinction à travers deux aspects (Kohlmann & Alt 2007):

- l'aspect métier, avec la modélisation des « **services métiers** », orchestrés en processus (Arsanjani 2004),
- l'aspect technique, avec la modélisation du système décomposé en « **services informatiques** » qui sont liés à des fonctionnalités logicielles à définir ou à réutiliser (Zhang et al. 2005).

Certaines méthodes de conception (ou re-conception) de services cherchent à couvrir ces deux aspects (Quartel et al. 2004), mais nombre d'entres elles ne se focalisent que sur l'un ou l'autre de ces aspects. C'est notamment le constat que fait (Kohlborn et al. 2009) après une analyse détaillée d'une trentaine de ces méthodes. Au-delà de cette analyse, il propose de définir une « approche consolidée » pour l'identification de nouveaux services « qui pourront être fournis aussi bien à un niveau métier qu'à un niveau technique ». Dans les paragraphes qui suivent, nous synthétisons son approche en l'illustrant par l'exemple d'un service de devis.

A partir du modèle du domaine et des exigences, les services métiers sont définis par :

- les ressources, à savoir le prestataire du service (ex. le responsable des devis), le contrat relatif au service, et les informations nécessaires à l'exécution du service (ex. la demande du client, la faisabilité, le devis...),
- le modèle interactionnel qui caractérise l'interaction avec le client, à savoir les tâches de celui-ci qui mènent à l'exécution du service et qui lui succèdent (ex. un service de devis est appelé par une requête du client, il est suivi par l'acceptation ou le refus du devis par le client),
- le modèle opérationnel qui décompose le service en un processus de sous-services (ex. recevoir la requête du client, clarifier la faisabilité, créer le devis).

Chaque sous-service est de nouveau décomposé en services atomiques (c.à.d. ayant le plus bas niveau de granularité), un service informatique étant alors assigné à chacun de ces services atomiques. L'analyse du système existant permettra d'identifier si un service informatique adapté existe déjà (c.à.d. est déjà implémenté et peut être réutilisé). Sinon il est nécessaire de le définir par :

- les fonctionnalités qu'il supporte, du type « créer », « modifier », « supprimer »,… (le développement par composants permettra de grouper ces fonctionnalités en composants et d'ainsi recomposer les services informatiques),
- les inputs et les outputs de chaque fonctionnalité.

Comme nous l'avons vu dans les approches IHM, il existe des approches visant à automatiser cette dérivation des services métiers en services informatiques. Par exemple, (Touzi et al. 2009) automatise la proposition de services collaboratifs dans une approche MDA à partir de modèles de processus collaboratifs. Il développe pour cela des règles de transformation pour traduire les modèles métier en modèle technique. La méthode est supportée par une suite d'outils. On retrouve le même genre d'approche dans (Bispo et al. 2010) ou encore (Delgado 2010).

Le développement de SI est donc supporté par la modélisation de processus qui définissent l'activité métier à un niveau organisationnel. Cependant, la modélisation des processus métier comporte certaines limites.

### *1.14.4 Les limites des approches basées sur le Business Process Modeling (BPM)*

Même si les modèles de processus peuvent servir de support au développement automatisé par des approches de MDA, il semble que dans un certain niveau de détail l'intervention d'un analyste et d'un développeur soit nécessaire. (Noran 2000) affirme que « la traduction d'un modèle métier en un modèle logiciel n'est pas un processus simple ». Pour (Soulier et al. 2007) « il parait illusoire de croire pouvoir se passer d'une phase d'implémentation ».

De plus, le BPM suscite des questions quant à sa capacité à représenter des situations collaboratives particulières. (Van der Aalst 1998) nous dira que « les processus collaboratifs sont hors de la portée de la définition des workflows. Dans un processus collaboratif, l'accent est mis sur la communication et le partage de l'information, plus que sur la définition d'un processus qu'il n'est alors pas nécessaire d'expliciter ». (Berard & Karlshoej 2012), citant (Curtis 1992) et (List 2006) fait ressortir le fait que les langages de modélisation de processus implémentent bien les aspects fonctionnels et comportementaux, c'est-à-dire les règles métiers et leur séquençage. Par contre, ils sont limités en termes de perspective organisationnelle (la modélisation des acteurs) et informationnelle (la modélisation des éléments d'information traités).

Cela s'avère d'autant plus vrai en qui concerne notre domaine d'application, le projet AIC, pour lequel la représentation sous la forme de processus métiers rigoureux tels que ceux utilisés dans la modélisation d'entreprise semble trop « rigide ». En effet, les équipes de projet de conception architecturale ne sont pas dirigées par une « couche stratégique » qui cherche à maximiser le rendement de l'entreprise par un processus optimisé. L'incertitude du processus décisionnel dans un projet AIC conduit les acteurs à devoir s'adapter à tout instant, tout en étant peu dans une logique de planification : c'est un processus dit « ad hoc » (Kubicki 2006). L'action est en effet étroitement liée à son contexte social et physique, plus qu'a une directive stratégique générale. (Suchman 1987) introduit le concept « d'action située ». On peut également lire dans (Hanser 2003) qu'"un projet architectural n'est jamais monotone car chaque nouveau projet correspond à un contexte particulier, il serait donc illusoire de tenter de dégager un processus précis et reproductible d'un projet à un autre ».

Tout comme le font (Sierhuis & Clancey 2002) ou (Soulier & Lewkowicz 2006) nous préférons parler de « pratiques ». En effet, le terme pratique est synonyme de méthode, de manière de faire mais aussi d'accomplissement : on parle de « pratique ingénieuse », ou de « bonne pratique ». La pratique se réfère également à une expérience, une habitude et à une amélioration continue (cf. « acquérir de la pratique »). On confère alors au concept de pratique une certaine souplesse qui favorise l'amélioration. Dans la caractérisation des besoins métiers relatifs au projet AIC, nous adoptons donc le concept de « pratique métier » plutôt que celui de processus pour caractériser l'activité collective et individuelle.

## 1.15 Conclusion : vers « des services adaptés aux pratiques métiers »

Comme on peut le lire dans (Emig & Weisser 2006), la conception orientée processus métier « est la prochaine étape dans l'évolution de la conception logicielle », « elle ne remplace pas mais complète les approches existantes ». Elle permet en effet de relier les aspects techniques aux besoins métiers de l'entreprise afin que ces derniers conditionnent le développement logiciel. En d'autres termes, elle apporte aux approches de GL la dimension « organisationnelle » de l'entreprise aux besoins métiers analysés, tout comme la conception IHM apportait la dimension « interactionnelle ». Dans les approches de conception orientées services, l'organisation est décomposée en **services métiers** qui sont supportés par des **services informatiques ICT**.

C'est ce que résume la figure suivante (Figure 52) en resituant ces différentes approches par rapport aux différentes « couches » d'une architecture d'entreprise. On y distingue également les différents modèles qui les supportent (bulles vertes).

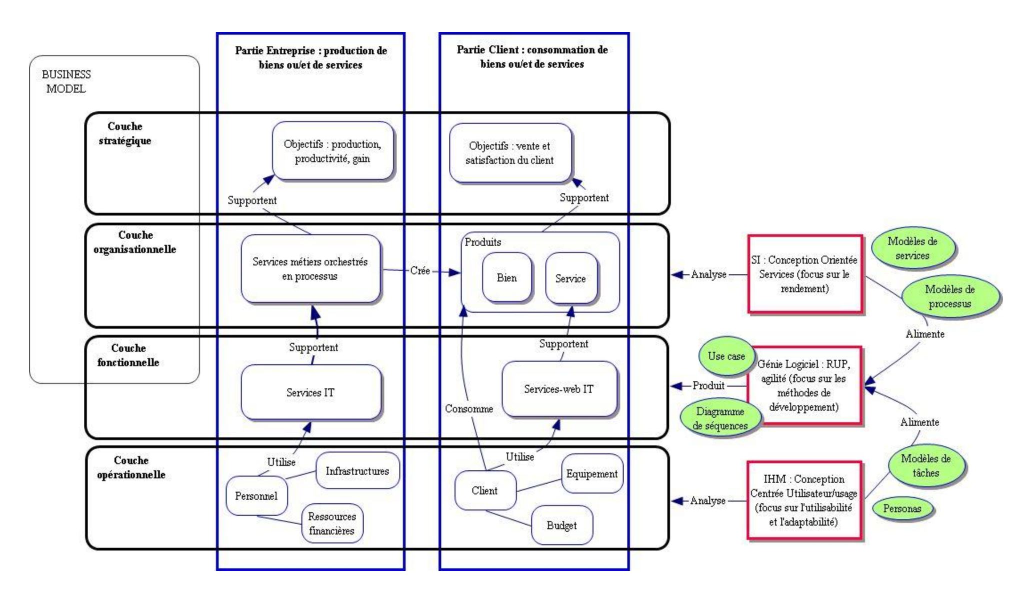



*Figure 52. Représentation des approches de conception de services ICT dans l'architecture de l'entreprise*

La modélisation des processus métiers est largement adoptée par les concepteurs de systèmes d'information dans le domaine de l'entreprise et ce à travers différents formalismes. Pourtant, elle semble comporter quelques limites quant à la modélisation de du contexte organisationnel d'un projet AIC. En effet, le travail de chacun est composé dans un cadre alternant supervision directe et ajustement mutuel : (Kubicki 2006) parle de « coordination flexible ». Or nous pensons, à la manière de (Marjanovic et al. 2007), que la communauté « BPM » est restée concentrée sur une coordination inflexible. Devant ces préoccupations, nous avons mené une première réflexion sur le concept de **pratique métier**.

Le chapitre suivant oriente notre étude vers le domaine du TCAO. Nous décrivons ces outils en introduisant le concept de « service collaboratif ». Nous considérons ensuite les grands principes de conception de collecticiels (et donc de services collaboratifs) dits adaptés puis nous adoptons un point de vue plus technique par la description de leur architecture logicielle.

# Chapitre 6 – Travail Collaboratif Assisté par Ordinateur et Services Collaboratifs

Ce chapitre introduit dans un premier temps les notions propres à la description des outils de Travail Collaboratif Assisté par Ordinateur (TCAO) et étend cette description aux services qu'ils proposent : les services collaboratifs. Pour (Vissers et al. 2004) les services collaboratifs sont « conçus pour supporter les groupes de personnes qui interagissent entre elles dans leurs tâches collaboratives ». Nous mettons ensuite en avant les raisons d'échec de ce genre de services et les principes pour y remédier. D'après (Veer 1996), la conception des collecticiels forme « une nouvelle branche dans les méthodes de conception ». Dans la continuité des propos tenus dans les chapitres précédents, nous considérons cette branche comme un « nouveau paradigme ».

## 1.16 Description des outils de TCAO et des services collaboratifs

### *1.16.1 Les dimensions fonctionnelles et spatio-temporelles*

Les outils de TCAO aussi appelés collecticiels, supportent les pratiques collectives de leurs utilisateurs de plusieurs manières : on dira qu'ils couvrent un ou plusieurs « espaces fonctionnels » (Ellis & Wainer 1994). Les trois espaces fonctionnels principaux qui composent ce qu'on appelle le « trèfle fonctionnel » (Salber et al. 1995; Piquet 2009) sont:

- l'espace de production, supportant l'action des acteurs sur l'information,
- l'espace de coordination, supportant la planification des activités,
- l'espace de communication, supportant l'échange d'information.

On parle aussi du modèle 3C (Gerosa & Pimentel 2006) pour Communication, Coordination et Coopération, où le terme Coopération est utilisé pour parler de la « production conjointe des membres d'un groupe au sein d'un espace partagé ». (David 2001) propose une évolution du trèfle fonctionnel, intégrant un nouvel espace de « conversation ». Cet espace comprend les outils permettant la communication entre les acteurs, mais ne produisant pas d'information persistante, au contraire de l'espace de communication (Figure 53).

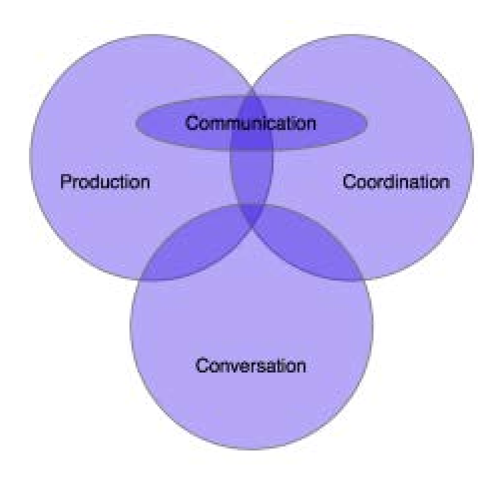



*Figure 53. Le modèle évolué du « trèfle fonctionnel »*

Lors de l'usage d'un collecticiel, les interventions des différents utilisateurs peuvent être conceptuellement réparties dans deux contextes différents : le contexte temporel et le contexte de localisation (Piquet 2009):

- on parle d'usages synchrones lorsque plusieurs utilisateurs agissent en même temps, et d'usages asynchrones lorsqu'ils interviennent à des moments différents,
- on distingue également les usages « dans un même lieu » des usages « à distance »

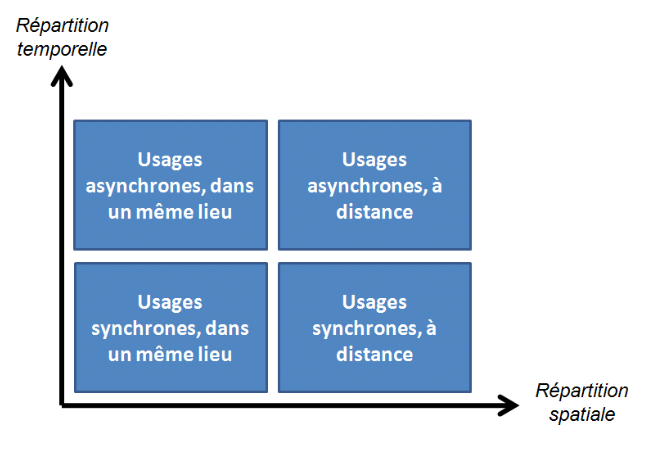



*Figure 54. Répartition des types d'usages dans le modèle spatio-temporel*

### *1.16.2 Les types de services collaboratifs*

Répartis dans les trois espaces fonctionnels et les quatre espaces spatio-temporels, les collecticiels offrent un certain nombre de ces « services collaboratifs » utilisés à différents moments de la collaboration (Laurillau 2002; Dewan 2001; Gerosa & Pimentel 2006), tels que :

- les services de gestion de l'emploi du temps,
- les services de partage et contrôle de l'information qui permettent de gérer l'accès des utilisateurs à l'information partagée (cela implique la consultation mais aussi la modification par un groupe comme la comparaison et la fusion des éditions, l'annulation…),
- les services de gestion d'interface qui permettent de propager les éléments graphiques entre les utilisateurs, de manière synchrone (tout le monde voit la même chose) ou de manière répartie (chacun a sa propre visualisation),

- les services de notification qui informent les différents utilisateurs des changements d'état du système,
- les services de courrier électronique, de tchat, de forum, etc…

Le schéma suivant recomposé à partir de (Laaroussi 2007) et (Guerriero 2009) illustre une répartition de ces services supportant la conception architecturale en fonction des dimensions fonctionnelles présentées précédemment (Figure 55). L'aspect réactif est relatif à une activité « qui évolue et s'adapte », « dont le contenu change avec l'environnement, avec la personnalité des acteurs ». Une activité prédictive doit quant à elle « être planifiée et instrumentée, où l'on définit au préalable les actions qui seront menées ».

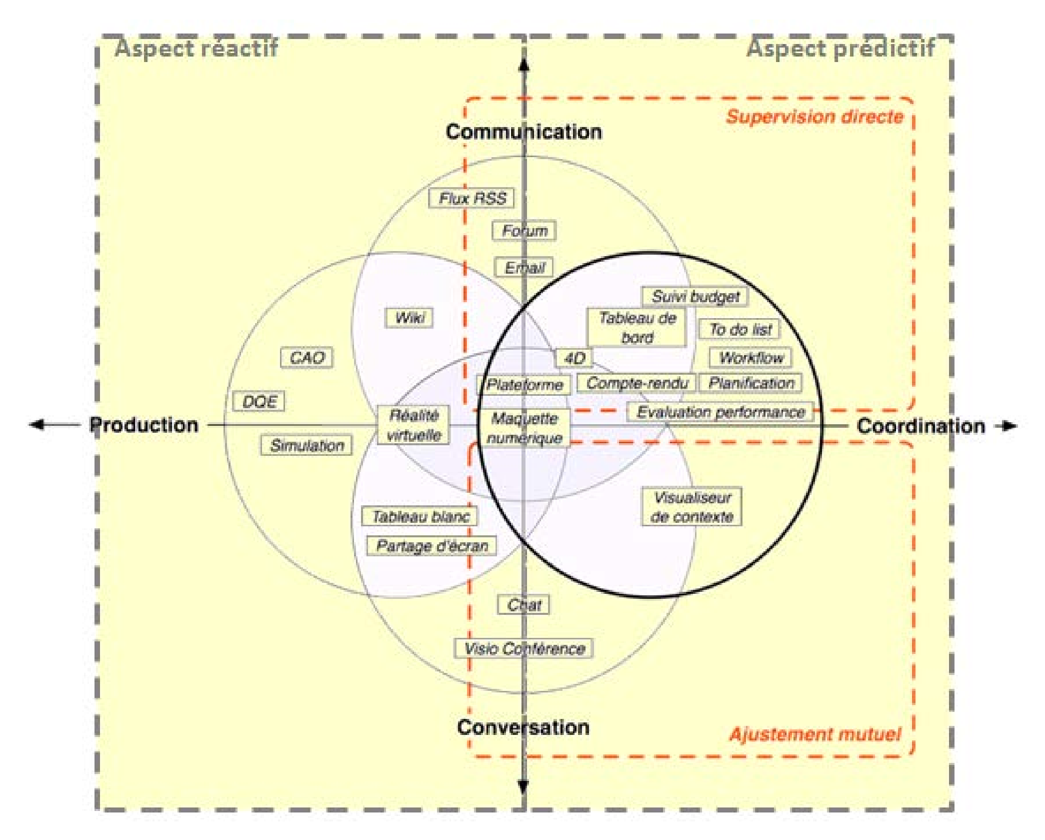



*Figure 55. Positionnement des services par rapport aux caractéristiques d'une activité collective ((Laaroussi 2007) et (Guerriero 2009))*

### *1.16.3 Description fonctionnelle*

Nous proposons l'étude de plusieurs modèles d'architecture logicielle spécifiques à la description de services collaboratifs. Ces modèles font évoluer le modèle MVC qui a retenu notre attention au § 1.9.3 (cf. Figure 26).

Le modèle Arch (L Bass et al. 1992) est une évolution directe du modèle MVC. Dans ce modèle un « adaptateur » abstrait les objets du domaine en objets conceptuels lors de l'interaction entre le Modèle (appelé Noyau fonctionnel) et le Contrôleur (appelé Contrôleur de dialogue). De même, les « objets d'interaction » de la Vue (le Composant Physique d'Interaction) sont abstraits en « objets de présentation » par un « Composant Logique d'Interaction ».

Le modèle Daewan introduit la prise en compte de plusieurs utilisateurs en faisant évoluer le modèle Arch. Chaque branche  est dupliquée pour chaque utilisateur et une partie partagée assure la mise en correspondance de ces branches. Les utilisateurs et les services communiquent entre eux à l'aide de deux types d'événements : les événements d'interaction reflétant l'interaction de l'utilisateur avec le service et les événements de la collaboration reflétant l'interaction entre les services  (Laurillau 2002).

Les modèles PAC-Amodeus et Clover (Laurillau & Nigay 2002) développent encore ces architectures, notamment en décomposant le cœur fonctionnel en agents (à savoir des programmes qui exécutent les requêtes). Nous ne souhaitons pas dans notre étude atteindre un niveau de description technique aussi avancé. Nous proposons au contraire de synthétiser ces informations pour décrire un modèle d'architecture simple que nous pourrons réutiliser pour décrire un service collaboratif.

Du modèle MVC au modèle Co-MVC

Nous proposons de repartir de l'architecture MVC que nous avons présentée précédemment (cf. § 1.9.3, Figure 26) et d'y ajouter la dimension collective introduite dans ce chapitre. Nous appelons ce modèle Co-MVC (Figure 56).

Comme l'illustre la Figure 56, le Modèle (ou Noyau Fonctionnel) est divisé en un modèle « partagé » (ou public) et un modèle « privé » répliqué pour chaque utilisateur. On comprend ici que lorsque le modèle du groupe effectue une requête ou se met à jour, cela se réplique chez les utilisateurs qui sont concernés. On parle alors de synchronisation.

Nous réintroduisons également les deux dimensions analysées au début de ce chapitre (en rouge sur la figure) :

- la dimension fonctionnelle, qui nous permet de distinguer les services de Communication, de Coordination et de Coopération
- la dimension spatio-temporelle qui permet de définir si des services agissent de manière synchrone ou asynchrone, colocalisée ou à distance.

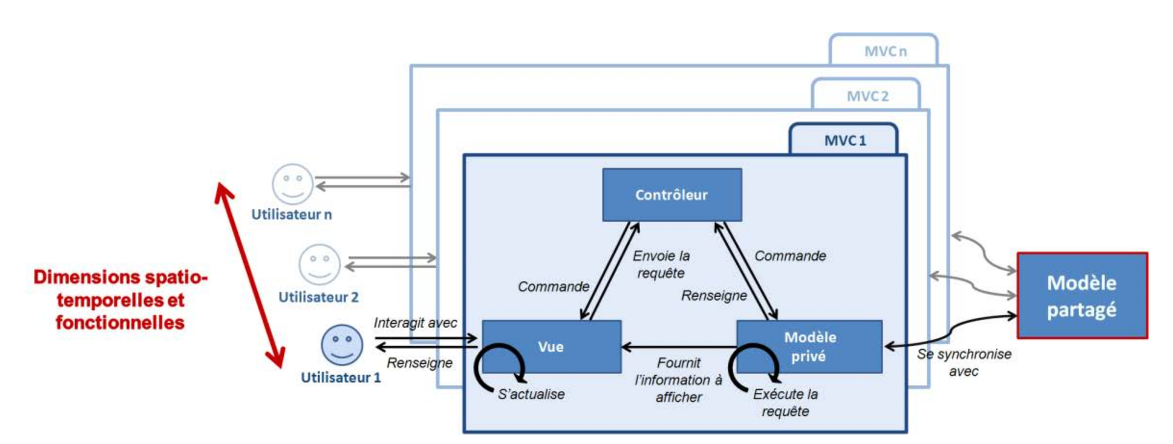



*Figure 56. Le modèle d'architecture Co-MVC pour décrire un service collaboratif*

## 1.17 Vers une réponse à « l'échec de la solution collecticiel »

### *1.17.1 Les constats d'échec*

Nous avons présenté, dans la première partie de ce manuscrit, l'expérience menée dans la conception d'un collecticiel particulier : CRTI-weB. Cet outil offre un certain nombre de « services collaboratifs » qui ont été définis pour le domaine particulier du projet AIC. Pourtant, nous avons pu identifier des manques d'adaptation de l'outil, entrainant certains freins à son utilisation par les professionnels du domaine. CRTI-weB n'est pas un cas isolé : la littérature nous montre que ces manques généraux d'adaptation trouvent leur source dans le processus de conception lui-même.

Jonathan Grudin (Grudin 1988; Grudin 1994) identifie trois cas d'échecs majeurs :

- l'outil lui-même échoue car il demande un travail supplémentaire à des utilisateurs qui n'en perçoivent pas directement le bénéfice,
- la conception de l'outil échoue car elle est basée sur des intuitions de bénéfices sans prévoir l'impact réel qu'aura l'outil,
- les concepteurs échouent car ils ne capitalisent pas d'expérience dans la conception, les outils étant trop complexes et ne permettant pas une analyse et une évaluation généralisable.

Selon (Greenberg 1991) l'échec vient du fait que les méthodes de conception logicielle ciblent les utilisateurs individuellement sans les considérer comme membres d'un groupe. L'insatisfaction de certains utilisateurs fait partie des limites « acceptables » dans le développement d'un outil logiciel classique. Or dans le cas d'un collecticiel, cette insatisfaction s'avère plus préjudiciable : elle a un impact négatif sur le travail de toute l'équipe, entrainant l'abandon général de l'outil. (Cockburn & Jones 1995) ajoute que les stratégies de conception « ne réussissent pas à prendre en compte les facteurs sociaux dans le travail de groupe ». Il relève également le manque de traçabilité des approches, les erreurs de conception étant sans cesse répétées.

Greenberg reste très critique quinze ans après dans (Greenberg 2006) : selon lui, les besoins et les possibilités sont grands concernant la suppression des barrières dues à la distance et l'augmentation du travail de groupe, mais l'adoption des collecticiels reste limitée (mis à part pour les services de messagerie instantanée qui sont maintenant largement implémentés). Il relève notamment la difficulté technique de développement des solutions, liée au manque d'outils pour la conception.

En ce qui concerne les projets AIC, (Nitithamyong & Skibniewski 2007) relève (au travers d'une étude sur 82 projets et 14 systèmes) que l'usage des collecticiels réduit les temps et les coûts induits par la collaboration mais « n'a pas d'impact significatif sur l'amélioration de la qualité du projet, la sécurité et la satisfaction du client ». (Kubicki et al. 2009) constate que les collecticiels :

- sont trop orientés « métier unique », à savoir qu'ils supportent l'activité d'un métier en particulier mais pas les activités d'un projet AIC dans ses dimensions coopératives,
- présentent un manque d'interopérabilité, de par la différence de format dans les données à traiter (particulièrement en ce qui concerne les représentations graphiques et la multitude d'outils utilisés à cet effet),

- ne représentent qu'une partie du contexte coopératif, c.à.d. de l'activité globale de projet, au travers d'informations « classiques » (comme le planning, les rapports, les listes de documents...)

### *1.17.2 Les principes d'un collecticiel adapté*

Devant les divers constats d'échec de la solution des collecticiels, (Greenberg 2006) relève l'intérêt de développer des solutions « personnalisables » qui s'adapteraient aux rôles de chacun. Il définit un « collecticiel personnalisable » comme un collecticiel dont le comportement peut être paramétré pour répondre aux besoins particuliers des participants d'un groupe ou du groupe lui-même. Le collecticiel personnalisable s'adapte d'un point de vue social (en prenant en compte les différences des utilisateurs au sein de leur groupe) et d'un point de vue technique (en permettant à l'utilisateur de choisir la solution qui répond le plus à ses besoins). Selon (Katsumata 2007), il déploie un mécanisme orienté par les tâches qui prédit les intentions et les demandes de l'utilisateur en fonction de sa situation dans le groupe et de son contexte d'utilisation.

(Cockburn & Jones 1995) présente un guide pour le développement d'outils support au travail collaboratif. Ici encore, l'utilisateur et son contexte (métier et technologique) sont mis au centre des préoccupations. Ce guide est composé de 4 principes :
- maximiser l'acceptation personnelle, c.à.d. encourager les individus à adopter le système en explicitant ce qu'il « leur apporterait »,
- minimiser le recours aux exigences de l'utilisateur et ainsi réduire la disparité entre les coûts et les bénéfices,
- minimiser les contraintes, c.à.d. laisser l'utilisateur libre de gérer son travail de la manière qui lui parait la plus adaptée et non pas comme le système lui impose,
- maximiser l'intégration externe, c.à.d. l'intégration dans l'environnement technologique actuel de l'utilisateur.

### *1.17.3 « Loose coupling » et conception de services collaboratifs*

Le terme « loose coupling » (littéralement « couplage lâche ou faible ») est utilisé pour décrire la relation entre des personnes ou groupes de personnes dans laquelle les évènements de chacun sont couplés et réactifs (c.à.d. aux actions des autres) mais préservent leur propre identité et leur propre contexte (Pinelle & Gutwin 2005). On distingue le « loose coupling » dans lequel les besoins de communication sont réduits et les personnes ont seulement « besoin d'être au courant de l'activité d'autrui, du « tight coupling » (couplage rigide ou fort) caractérisé par une communication, une dépendance et une interaction fortes entre personnes (et/ou groupes) (Olson 1996; Grinter et al. 1999).

Le concept de « loose coupling » s'applique particulièrement aux organisations composées d'équipes multidisciplinaires dans lesquelles chaque acteur possède une certaine autonomie, sa propre expertise et les pratiques qui lui sont liées, comme c'est le cas pour les équipes de projet AIC.

(Pinelle & Gutwin 2005) présente une suite d'approches de conception de collecticiel supportant un travail collaboratif basé sur ce concept de « loose coopling » (Tableau 9). Ces approches sont générales, s'appliquant au développement de tous types de services. On pourra trouver dans (Sohlenkamp et al. 2000) ou (Sutcliffe 2005) des séries d'exigences plus spécifiques relatives à la conception d'un service de notification.

Tableau 9. Résumé des approches de conception basée sur le concept de « loose coupling »

| Approches de conception | Descriptions |
|---|---|
| Support de l'autonomie et de la flexibilité | Support des pratiques actuelles de travail, sans forcer l'interdépendance des acteurs au risque de réduire leur autonomie et leur flexibilité. |
| Agrégation de l'information | Déplacement de parties d'information depuis des espaces tampons vers un unique repository pour améliorer la coordination et la conscience des activités du monde réel |
| Support des espaces de travail individuels et du partage discret | Le partage d'information avec un ou plusieurs membres de l'équipe doit pouvoir se faire à la discrétion des autres acteurs |
| Intégration de la collaboration avec des équipements du travail individuel | Les équipements collaboratifs ne doivent pas interférer avec l'utilisation d'autres équipements plus généralement utilisés |
| Facilité de la conscience asynchrone | Support de la conscience des activités des autres et persistance de l'information avec le temps |
| Support de la coordination « lâche » | Minimiser les efforts de coordination et la négociation directe des utilisateurs |
| Support de la communication « lâche » | Minimiser les efforts requis pour initier une communication |
| Support du changement vers un couplage plus « rigide » | Supporter des interactions et communications directes ou du moins en supporter l'organisation |
| Préservation d'une organisation flexible | Laisser les acteurs déterminer leur niveau d'implication dans les situations collaboratives |

## 1.18 Conclusion : « concevoir des services collaboratifs »

Les études sur le TCAO (couplées à notre expérience dans le domaine avec l'outil CRTI-weB) nous ont permis d'identifier un certain nombre de **« services collaboratifs »** qui composent un collecticiel. Ces services supportent la communication, la coopération et la coordination et ceci dans des contextes de localisation et temporel différents. Nous avons défini un modèle d'architecture logicielle qui décrit un service collaboratif d'un **point de vue fonctionnel**.

Pour pallier aux manques d'adaptation des collecticiels, les approches de conception se sont tournées vers l'analyse de grand principes ou caractéristiques propres au travail collaboratif. Nous pouvons assumer à partir des quelques exemples donnés que la clé de cette adaptation réside dans le respect de l'utilisateur, c.à.d. la capacité du collecticiel à respecter les pratiques de travail (et la discrétion) des utilisateurs sans les altérer, ainsi que l'efficacité, c.à.d. la capacité à réduire les efforts induits par certaines tâches. Ces principes ou approches ne restent pourtant que des lignes directrices sans être vraiment des spécifications pour la conception de services collaboratifs adaptés.

Un nouveau paradigme

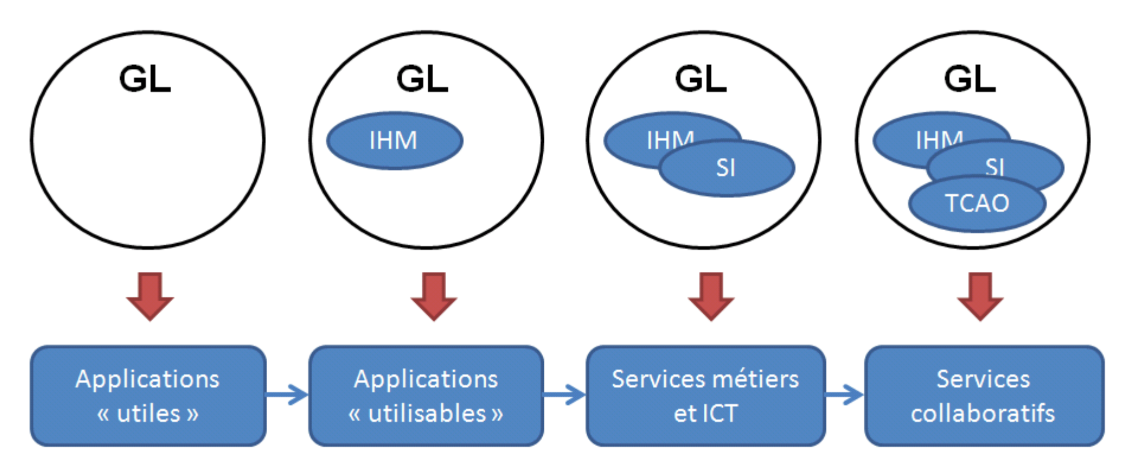



*Figure 57. Vers un nouveau changement de paradigme dans la conception logicielle*

Au terme des analyses menées dans les chapitres 4 et 5, nous avons pu observer comment le domaine des IHM et des SI « alimentaient » le Génie Logiciel pour concevoir respectivement des Interfaces adaptées aux utilisateurs et des Services Informatiques répondant aux besoins d'une organisation. La conception de services collaboratifs étudiée dans ce chapitre 6 peut être vue comme une « nouvelle branche », un troisième paradigme de la conception logicielle (Figure 57). Le tableau ci-dessous résume les avantages et les limites des différentes approches (Tableau 10).

Tableau 10. Avantages et limites des champs étudiés

| Domaines | Avantages | Limites |
|---|---|---|
| **GL** | Méthodes structurées basées sur des modèles<br>Séparation de l'analyse métier et de la spécification du système | Pas de prise en compte du contexte de l'utilisateur<br>Méthodes orientées exigences et non utilisabilité |
| **IHM** | Prise en compte approfondie de l'utilisateur, de ses préférences et de son contexte<br>Tâches interactives clairement définies | Point de vue mono-utilisateur |
| **SI** | Point de vue organisationnel<br>Prise en compte des besoins de l'entreprise, accent sur l'efficacité et la productivité<br>Réutilisation de services existants | Processus rigides guidés par une stratégie, peu adaptés à des situations collaboratives particulières<br>L'individu est mis de côté |
| **Collecticiels** | Basés sur des grands principes de la collaboration<br>Points de vue aussi bien individuels que collectifs | Les principes de collaboration ne sont pas spécifiques à un domaine<br>Les approches sont trop théoriques et manquent de méthode |

Au travers des différentes analyses nous pouvons conclure que :

- un collecticiel doit fournir des services adaptés à chacun des utilisateurs, en tant qu'individus mais aussi en tant que membres d'un groupe de travail et éléments du contexte coopératif dans lequel ils sont impliqués,
- la conception des services collaboratifs se doit d'être méthodique et outillée, au regard des méthodes classiques de conception logicielle, tout en prenant en compte les grands principes de la conception de services collaboratifs mais aussi les spécificités fonctionnelles de ce type de services.

Le chapitre suivant propose une description de trois méthodes particulières de conception de services collaboratifs pour assister des besoins métiers. Ces études nous ont permis de dégager les points d'intérêt et les limites de telles approches afin de construire notre propre méthode.

# Chapitre 7 – Les méthodes de conception de services : études de cas

Ce chapitre développe trois méthodes de conception de services particulières. L'objectif est de montrer l'intérêt et la manière d'utiliser des modèles pour mener un projet de développement de services adaptés. Les bases de notre proposition sont définies en fin de chapitre, en mettant en évidence les points d'intérêt et les limites de ces trois méthodes. La première illustre particulièrement bien l'utilisation d'outils de modélisation dédiés pour éditer les différents modèles et le passage d'un modèle à l'autre. La deuxième met en avant la correspondance entre modèles et leurs transformations dans un cadre de référence d'ingénierie dirigée par les modèles. Enfin, la troisième est une méthode dédiée au projet AIC, notamment par la spécification d'objets de modélisation du bâtiment.

## 1.19 CoCSys, une méthode de conception de collecticiels dirigée par des modèles

Dans sa thèse, Olivier Delotte (Delotte 2006) présente CoCSys, un processus d'assistance à la construction et l'évolution d'applications mobiles « en s'appuyant sur une architecture et des mécanismes à base de modèles ». Il s'agit d'un processus itératif basé sur la transformation de modèles (voir la Figure 58) :

- la phase 0 est la phase d'identification des scénarios d'utilisation donnant lieu à création d'un modèle de scénarios,
- la phase 1 consiste à créer un modèle comportemental sur la base des scénarios,
- la phase 2 transforme le modèle comportemental en modèle d'architecture logicielle,
- la phase 3 supporte l'itération par la prise en compte des nouveaux scénarios d'utilisation dans le modèle comportemental.

Cette section en présente dans un premier temps le principe de fonctionnement théorique puis une analyse critique qui questionne notamment l'applicabilité de cette méthode.

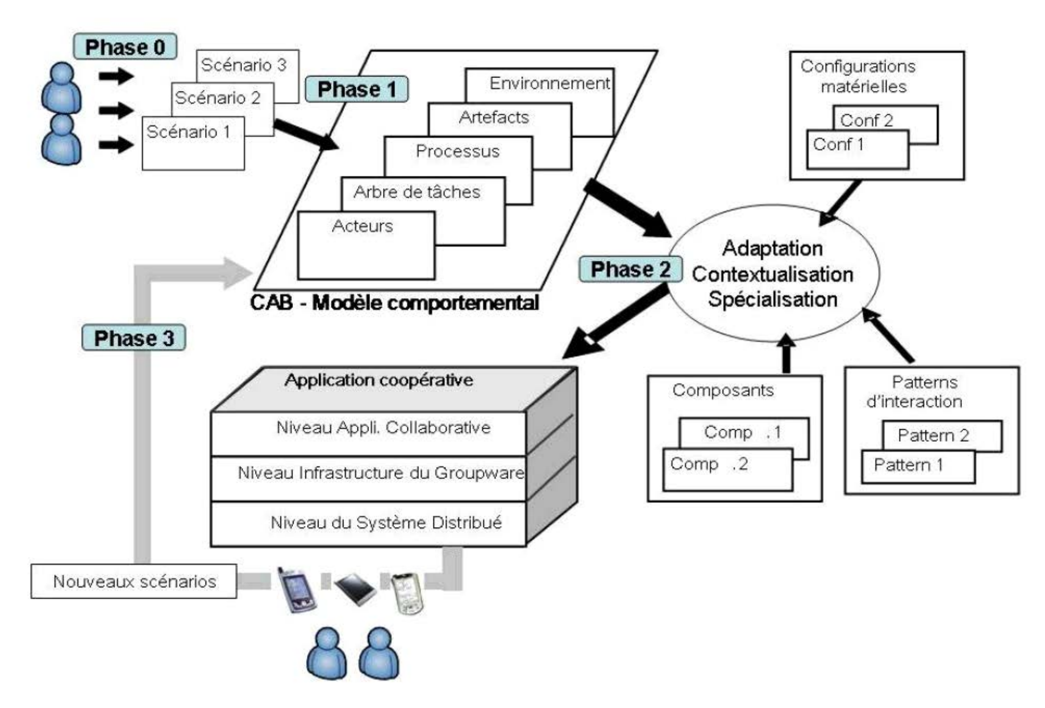



*Figure 58. Schéma du processus CoCSys (tiré de (Delotte2006))*

### *1.19.1 Déroulement de la méthode*

La formalisation des scénarios

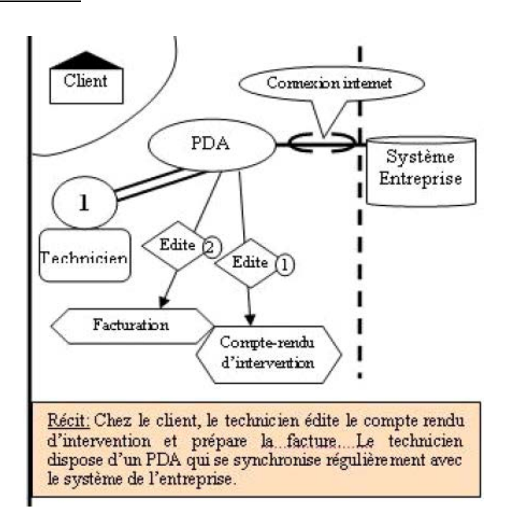



*Figure 59. Un exemple de scénario contextualisé dans son éditeur*

Le scénario est dans cette approche une vision partielle des besoins des utilisateurs dans un certain contexte. Afin de formaliser ces « scénarios contextualisés », les auteurs ont créé un formalisme de modélisation hybride supportant une description textuelle (sous forme d'histoires courtes) et une description graphique proche du cas d'utilisation (voir Figure 59). Ce modèle est l'instanciation d'un méta-modèle, le MMSC (méta-modèle du scénario contextualisé) il est édité par l'outil CBME (Contextual Behaviour Model Environment, Figure 65). Voici les concepts définis par le MMSC et leurs représentations respectives avec le formalisme de CBME (Figure 60, Figure 61, Figure 62, Figure 63, Figure 64) :

- Un acteur est défini par son rôle dans l'activité, et par un identifiant qui permet de le définir comme (1) l'acteur principal qui réalise les tâches décrites, (2) un acteur secondaire impliqué mais non responsable ou (-) un acteur passif qui est présent mais non concerné par l'action. Chaque acteur et le système possèdent leur propre zone de définition sur le graphique (appelée « espace de vie ») dont les limites sont représentées par des lignes en pointillés.

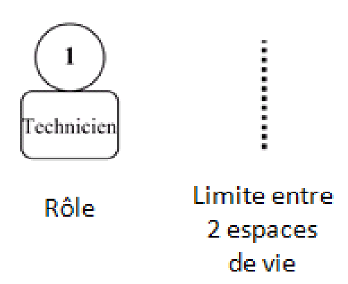



*Figure 60. Représentation d'un acteur et d'une limite entre deux « espaces de vie » dans CBME*

- Le contexte géographique et le contexte temporel localisent l'acteur dans l'espace et dans le temps.

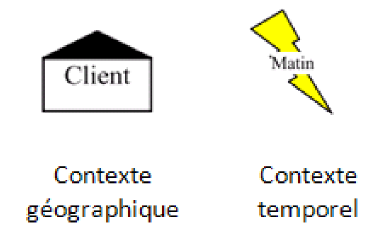



*Figure 61. Représentation des contextes géographique et temporel dans CBME*

- L'artefact est un objet créé, manipulé ou détruit par une tâche, tandis que le dispositif et le système sont respectivement le support interactif et le logiciel utilisés




*Figure 62. Représentation de l'artefact, du dispositif et du système dans CBME*

- Les relations sont de type Possède, Accède, Contacte, Synchronisation de données, Exécute (une tâche), Ordonnancement de tâche

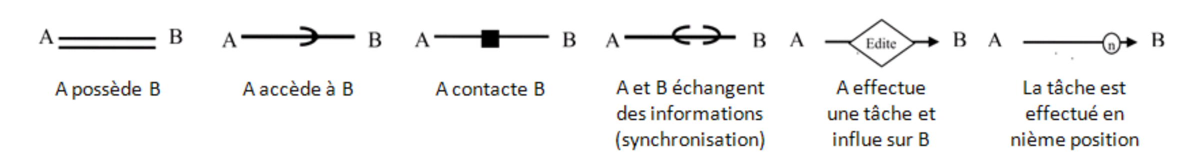



*Figure 63. Représentations des relations du scénario contextualisé dans CBME*

- Il existe enfin deux types de pré-conditions : optionnelles ou obligatoires

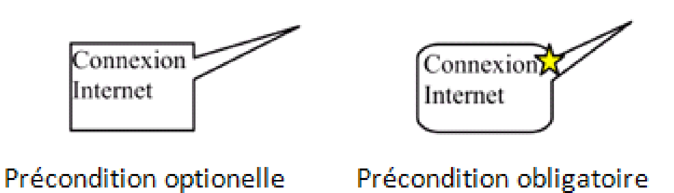



*Figure 64. Représentation des deux types de pré-condition du scénario contextualisé dans CBME*

Les scénarios contextualisés sont exprimés de façon textuelle par les utilisateurs puis interprétés et formalisés graphiquement par les concepteurs qui en dégagent un objectif et peuvent également proposer des scénarios alternatifs. L'étude ne donne pas de retours quant à l'écriture de ces modèles, que ce soit par rapport à l'utilisabilité de l'outil ou la compréhension des concepts. D'un point de vue externe, et sur la base des éléments illustrés, les éléments utilisés semblent clairs même s'ils gagneraient peut-être à être davantage distingués graphiquement. Ils ne reflètent par contre pas beaucoup le domaine d'application du fait du peu d'information donnée pour chacun d'eux (quel type de technicien est-ce ? Quels sont les attributs de la facture ? etc…).

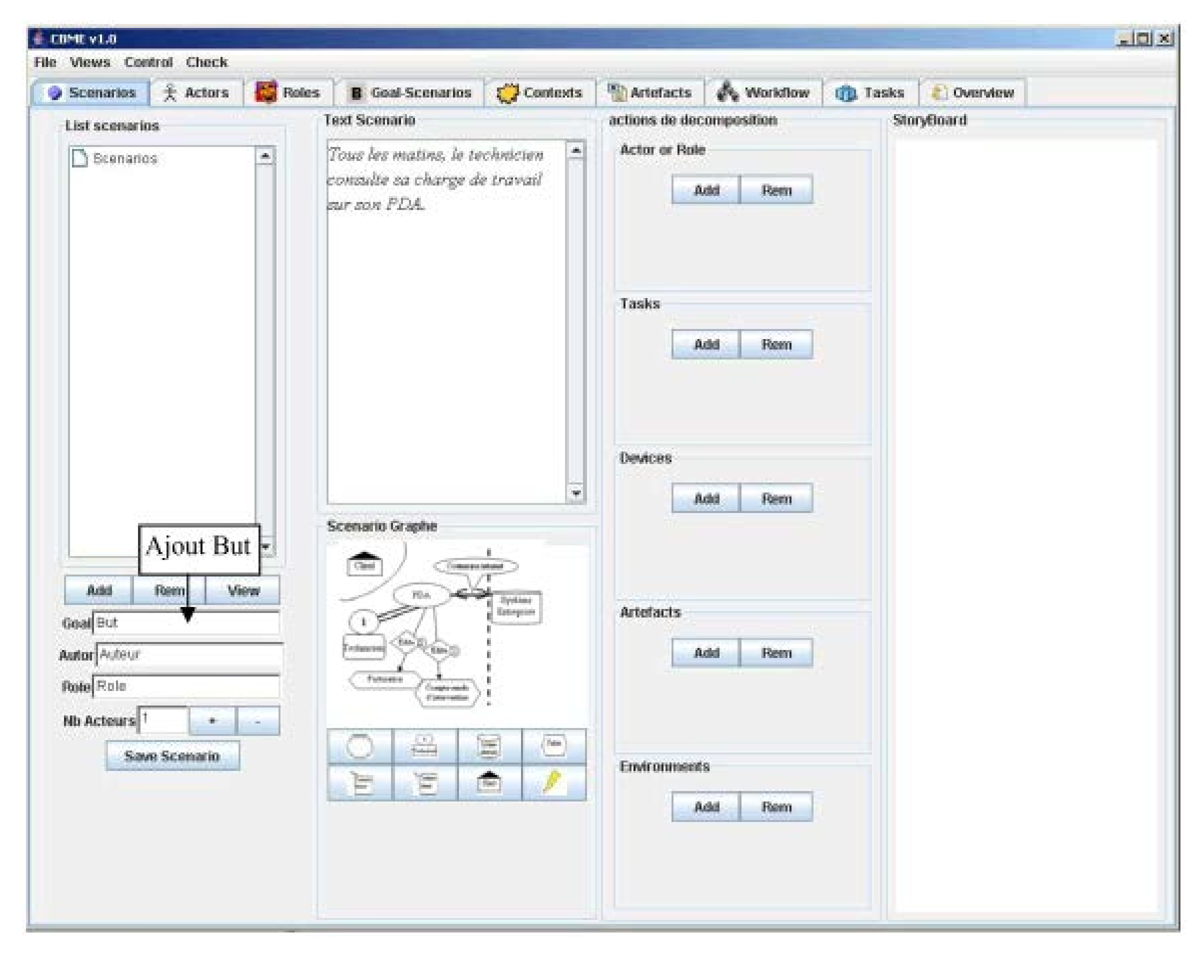



*Figure 65. Utilisation de CBME pour écrire les scénarios contextualisés (tiré de [Delotte2006])*

Le modèle comportemental CAB (Contextual Application Behavior)

Le modèle comportemental CAB a pour but de synthétiser les besoins des utilisateurs. Il est créé à partir des modèles de scénarios. Ce modèle doit être utilisable pour l'intégration des services et outils nécessaires à la réalisation des tâches. Le CAB est en réalité un ensemble de modèles ayant chacun leur propre formalisme :

- Le workflow basé sur le framework « 2flow » (Saïkali 2001) qui s'appuie sur un méta-modèle du workflow flexible (MMWf).

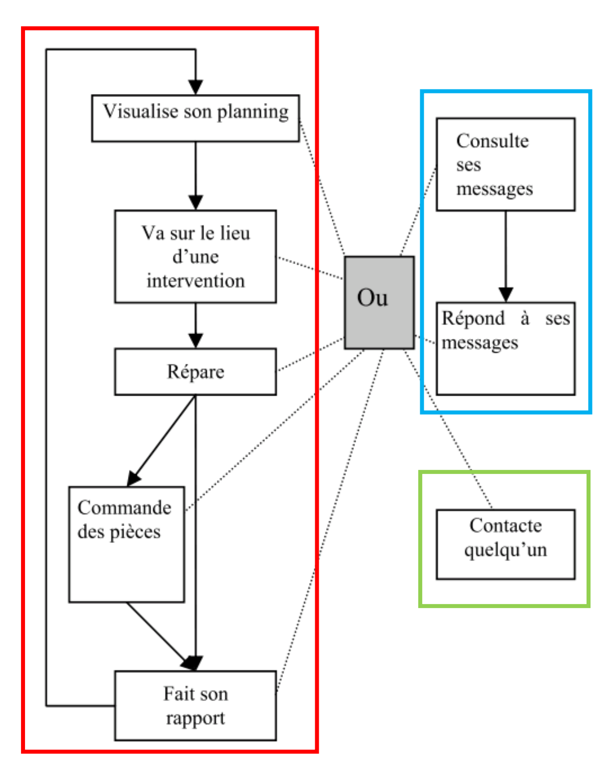



*Figure 66. Exemple de représentation de trois processus alternatifs (encadrés colorés) avec le formalisme « 2flow »*

- Le modèle de tâches, instanciation d'un méta-modèle de tâches (MMT) est exprimé avec le formalisme CTT (Mori et al. 2002) (cf. § 1.10.3) augmenté d'une catégorisation des tâches relative au trèfle fonctionnel (les tâches de Production, de Communication et d'Organisation) (Piquet 2009). Les arbres de tâches du CAB sont composés de tâches abstraites qui seront spécifiées ensuite en fonction de la configuration matérielle et du dispositif d'interaction.
- Le modèle du contexte, instanciation d'un méta-modèle du contexte (MMC), qui décrit l'environnement de l'utilisateur composé de « l'ensemble des informations qui ont une influence sur les besoins et les capacités de l'utilisateur à communiquer » et des dispositifs qu'il utilise.

Le méta-modèle comportemental est une agrégation de ces 3 méta-modèles (MMWf, MMT, MMC) et décrit aussi les modèles de rôle et d'artefact.

La création de l'application

Pour l'application de cette méthode, (Delotte 2006) décompose l'architecture logicielle d'un système coopératif en trois niveaux :

- Le CUO-M (Collaborative User-Oriented Model) supervise les interactions et propose les interfaces de présentation. Il est décrit dans le formalisme AMF-C, une évolution du modèle

d'architecture logicielle multi-agents AMF (Agent Multi-Facettes) (Masserey & Samaan 2006)

- Le CSA-M (Collaborative System Architecture Model) contrôle les sessions, les utilisateurs et les groupes.
- Le DSI-M (Distributed System Infrastructure Model) distribue les messages et contrôle le contenu.

La deuxième phase de transformation assure le passage de modèle comportemental CAB au modèle d'architecture logicielle. Les patrons de tâches, les composants logiciels et les patrons d'interaction forment des catalogues dans lesquels vont s'alimenter chacune des trois étapes de cette phase 2.

- Étape 1 : les patrons de tâches permettent de choisir dans une bibliothèque, le déroulement souhaité de certaines tâches utilisateurs en une suite prédéterminée de tâches concrètes (d'interaction et système) dépendante du contexte.
- Étape 2 : « les composants sont les éléments logiciels disponibles pour être intégrés lors de l'adaptation ». Si un composant n'existe pas, une demande de développement est formulée. Ces composants peuvent être des applications spécifiques « dont la portabilité doit être gérée en amont ». Le processus KMDE (Knowledge-Model Driven Engineering) réalise le passage entre le workflow du modèle CAB et son implémentation, par l'identification et l'intégration des composants fonctionnels souhaités. L'outil KMDEg supporte ce passage. Il présélectionne les services susceptibles de répondre aux besoins identifiés et permet au concepteur de valider ou modifier le choix.
- Étape 3 : la figure suivante (Figure 67) illustre la génération de l'application par la migration des services formalisés en AMF vers le dispositif.

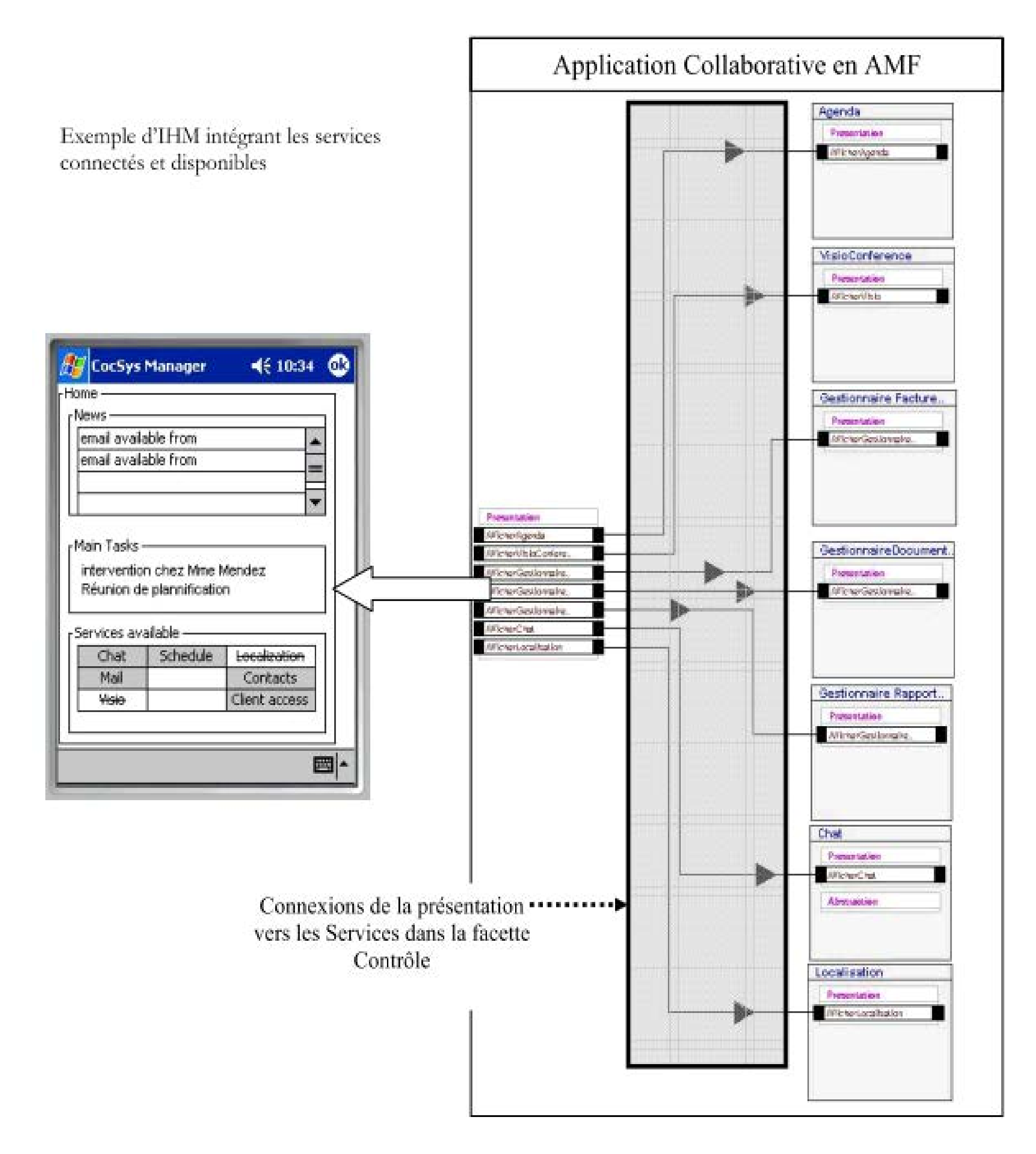



*Figure 67. Migration des services vers le dispositif*

### *1.19.2 Analyse critique*

La méthode de conception CoCSys est basée sur les approches MDA de transformation de modèles et se déroule en plusieurs phases : l'identification d'un scénario contextuel (décrivant des cas d'utilisation), la spécification d'un modèle comportemental puis le développement d'une architecture logicielle. Elle supporte la prise en compte des besoins relatifs au rôle d'une personne dans son organisation pour lui proposer des services informatiques adaptés. La méthode permet en plus de considérer le contexte utilisateur de cette personne. L'auteur propose une méthode de conception d'applications, à savoir un processus de conception à suivre ainsi que les formalismes et les outils de modélisation supportant ce processus (notamment CBME et KMDEg). **Le but de cette méthode est de générer une interface qui permet d'accéder à des services qui ont été identifiés comme nécessaires à l'exécution d'un processus métier.**

Le « méta-modèle des scénarios contextualisés » (MMSC) décrit les scénarios métier avec des concepts particuliers. Nous constatons que ce méta-modèle est très général et qu'il n'entre pas dans le détail des éléments décrits (quel type de technicien est-ce ? Quels sont les attributs de la facture ? etc…). Si cela peut suffire dans un contexte professionnel simple, nous pensons qu'un tel méta-modèle devra apporter plus d'éléments de description pour un contexte plus complexe comme celui du projet AIC. Le formalisme de modélisation des scénarios est à la fois graphique et textuel. **L'approche par création d'un méta-modèle et d'un formalisme dédié nous semble par ailleurs intéressante**. En ce qui concerne l'utilisation de ce formalisme, nous notons que les utilisateurs eux-mêmes n'utilisent pas l'éditeur de scénarios afin d'exprimer leur besoin. La description de leurs pratiques habituelles est retranscrite a posteriori par les concepteurs.

La méthode est une méthode de conception d'outils collaboratifs, pourtant la description des scénarios est toujours individuelle. Elle n'est pas rattachée à un comportement, un besoin collaboratif plus général qui implique la mise en place de ces scénarios individuels. Dans le cas d'étude présenté (une société de dépannage), les processus de travail sont établis et figés. Il n'est pas nécessaire de retracer les objectifs généraux. Mais dans un contexte plus complexe, on peut pressentir qu'**il faudra chercher davantage à comprendre comment se déroule la collaboration afin d'identifier les pratiques de chacun**, les unes vis-à-vis des autres. Cela permettrait de mieux décrire ces pratiques mais aussi de pouvoir les optimiser dans une problématique d'amélioration de la collaboration.

Nous pouvons nous poser la question de la séparation entre ce qui relève de la pratique métier (autrement dit ce qui doit être fait par des professionnels, quel qu'en soit le moyen) et ce qui relève de l'usage du dispositif pour assister cette pratique métier. Dans l'exemple proposé, la différence apparait clairement : la phrase « le technicien édite le compte rendu et prépare la facture » décrit une pratique liée au rôle et à la mission du technicien, alors que des informations telles que « tous les matins », « chez le client » ou « avec un PDA » sont des informations contextuelles relatives à un usage particulier. Dans ce contexte métier, les scénarios de travail et les usages qui les assistent sont répétitifs et prédéfinis. Aussi, il n'est pas réellement nécessaire de séparer leur modélisation. Ici encore, dans un contexte plus complexe comme celui du projet AIC, nous pressentons la nécessité d'aller plus loin dans la modélisation, en identifiant des variations d'usage pour une même pratique (sur le chantier avec un smartphone et au bureau avec un ordinateur par exemple). Comme nous pouvons le lire (p.149): « il est nécessaire de reformuler certains éléments du récit pour avoir une homogénéisation lexicale (processus de transformation) ». Par exemple, le verbe « recevoir » dans le premier scénario devient « télécharger ». Nous retrouvons ici au travers des propos de l'auteur, la **nécessité d'opérer une transformation entre une description « métier » (recevoir) et relative à un outil (télécharger).** Nous pensons qu'il est nécessaire de traiter deux modèles différents plutôt que d'adapter le même modèle.

La méthode CoCsys propose des phases de transformation structurées qui, de l'analyse des besoins à la spécification de l'architecture de l'outil, permettent de justifier chaque étape en retraçant le processus de développement. Pourtant, le passage des scénarios au modèle comportemental n'est pas systématique. Les modèles de scénario contextualisé et comportemental sont instanciés à partir de méta-modèles (le MMSC et le MMC). Cependant, on ne distingue pas le lien entre ces deux méta-modèles. Il aurait été intéressant de pouvoir mettre en correspondance le MMSC au MMC pour formaliser le passage d'un modèle un autre.

En ce qui concerne la deuxième phase de transformation, son applicabilité dans divers cas n'est pas vérifiée dans la thèse analysée. Des travaux scientifiques ultérieurs à la thèse analysée comme (Yin 2010) ou (Bain et al. 2009) réutilisent la méthode CoCsys comme base de leurs travaux scientifiques. Cependant, la réelle applicabilité de la méthode dans un contexte professionnel n'est pas évaluée et montrerait peut-être ses limites. Nos expériences dans le secteur AIC et notamment l'expérience CRTI-weB (cf. Chapitre 2) soulignent le risque de proposer des services inadaptés malgré un processus de conception guidé par l'analyse métier.

Du fait que cette méthode a été pensée pour la conception de systèmes mobiles et pour des processus métiers figés, elle trouvera certainement ses limites dans d'autres contextes professionnels plus complexes comme le Contexte Coopératif d'un Projet de Construction (voir § 1.1). Cependant, en nous inspirant de CoCsys, nous pouvons déjà identifier des objectifs relatifs à la conception de notre propre méthode qui puisse être fonctionnelle dans ce contexte. Nous souhaitons :

- analyser les pratiques existantes en utilisant des concepts et formalismes adaptés au domaine,
- assurer le passage de l'analyse métier à la proposition de services par la mise en correspondance de méta-modèles les définissant,
- supporter le développement de nouveaux services (ou leur amélioration) dans des contextes d'usage différents mais répondant aux pratiques analysées.

## 1.20 La méthode Symphony et les modèles pour la collaboration

Nous avons évoqué, plus tôt dans ce manuscrit (cf. § 1.12), le point de vue de Sophie Dupuy-Chessa (Dupuy-Chessa 2011) quant à la croisée des chemins entre la modélisation des Interfaces Homme-Machine (IHM) et des Systèmes d'informations (SI) pour la conception logicielle. Nous présentons ici la méthode de développement proposée par son équipe et basée sur la spécification d'exigences à la fois organisationnelles et interactionnelles. Nous nous intéressons tout particulièrement aux activités collaboratives mises en place au cours de ce processus.

### *1.20.1 Symphony et symphony étendue*

Symphony est une méthode de développement de système interactif itérative et incrémentale. Un système interactif est composé de deux parties : l'interface utilisateur et le noyau fonctionnel. L'interface utilisateur contient les éléments logiciels et matériels dédiés à la capture des entrées de l'utilisateur et à la restitution des sorties du système. Le noyau fonctionnel contient le reste du système, c'est-à-dire ses composants de calcul et de stockage de l'information. La méthode Symphony est une méthode basée sur l'utilisation de composants (cf. § 1.9.1) appelés objets. Les « objets interactionnels » décrivent l'interface, et sont spécifiés à partir des « objets métiers » qui eux décrivent le noyau fonctionnel.

Le but est de faciliter la collaboration entre les acteurs d'un projet de conception logicielle (particulièrement entre ceux de l'IHM et du GL) et de permettre aux concepteurs de produire des modèles cohérents, assurant ainsi une certaine qualité du produit. Quatre principes sont à respecter :

- « ne pas bouleverser les pratiques des acteurs » d'un projet de développement, ce qui implique leur laisser leurs outils et leurs modèles,
- permettre la synchronisation de ces acteurs sur des objectifs ou des modèles communs
- assurer « la traçabilité et la cohérence entre les modèles »,

- « fournir des outils de support à la méthode ».

Chaque itération dans l'application de la méthode donne la priorité à un processus métier et suit un cycle en Y (Figure 68). « La branche fonctionnelle de gauche correspond à la traditionnelle modélisation des besoins » du domaine (à travers des diagrammes de cas d'utilisation et des scénarios) et des utilisateurs (diagrammes de tâches, modalités d'interactions et maquettes). « La branche technique de droite permet aux concepteurs de concevoir les architectures techniques et logicielles » (Dupuy-Chessa et al. 2010).

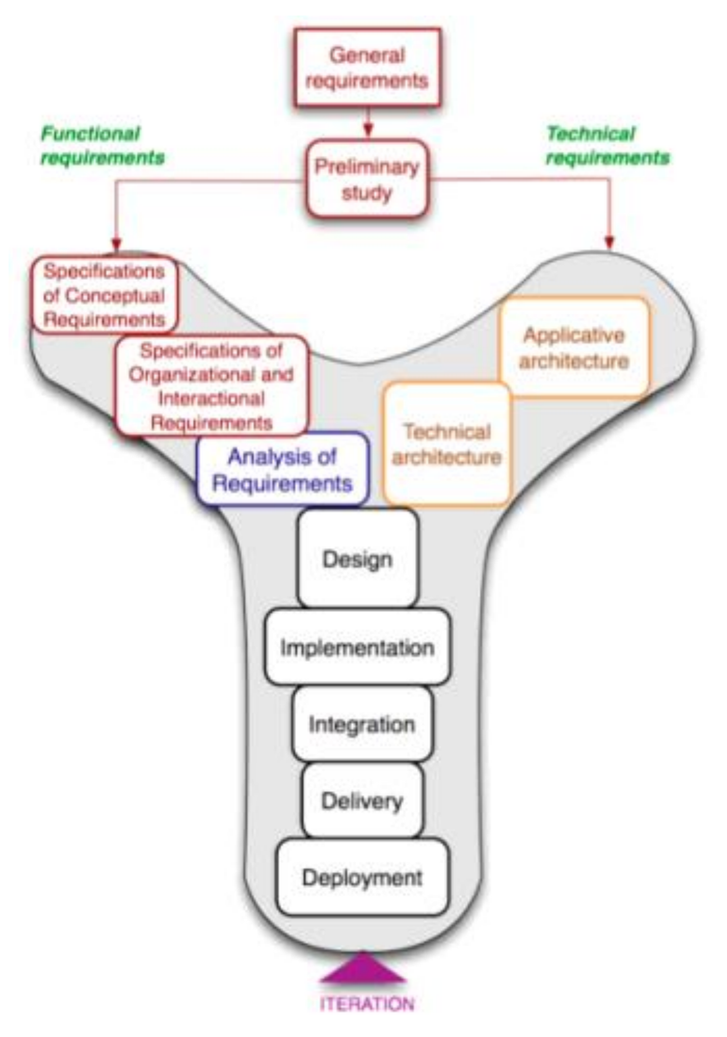



*Figure 68. Cycle de Symphony étendue*

L'extension « Symphonie étendue » complète cette approche « pour y inclure les pratiques de conception des systèmes de réalité mixte » (comprenant les systèmes de réalité augmentée et les interfaces tangibles). Elle propose ainsi une méthode de développement complète et appliquée à un contexte technique particulier.

Nous nous sommes particulièrement concentrés sur la branche fonctionnelle (branche gauche) en cherchant à identifier comment elle allie des préoccupations à la fois logicielles et interactionnelles pour répondre à des besoins métiers. La branche technique nous apporte en outre des éléments de description du contexte utilisateur.

Une phase préliminaire a pour but d’analyser le métier tel qu’il est pratiqué par le client, demandeur d’une solution logicielle, afin d’identifier les processus métier et leurs participants. Ces processus sont composés d’activités déclenchées par un input ou un évènement. La méthode Symphony propose une description de ces processus sous forme de scénarios en langage naturel. Cette phase est la responsabilité des experts métier et des experts en utilisabilité qui collaborent pour « capturer les prescriptions basées sur l’implémentation actuelle du processus métier et pour la définition d’un cadre de référence de l’utilisateur de l’application ».

Dans le cas d’application illustré dans (Dupuy-Chessa et al. 2010), l’objectif est de développer un service d’identification de dégâts à travers un système de réalité mixte (illustration Figure 69).

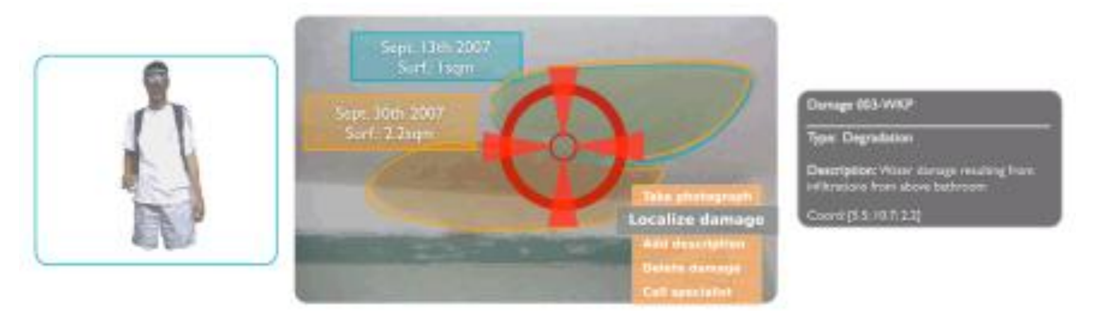



*Figure 69. Prototypage de l’identification d’un dégât à travers un système de réalité mixte*

### *1.20.2 La branche fonctionnelle*

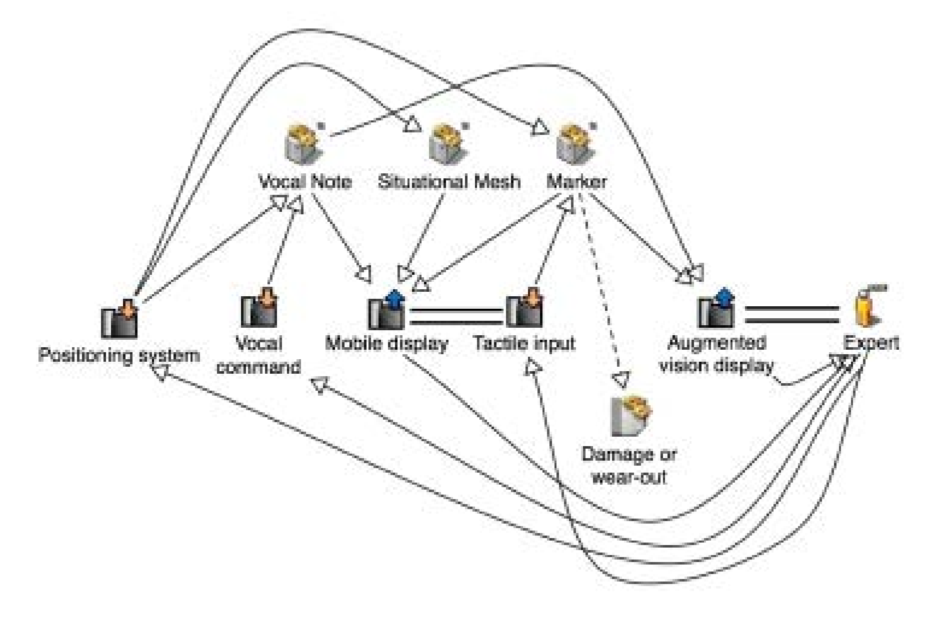
La spécification conceptuelle des besoins

Les processus métier et les acteurs qui y interviennent sont ici détaillés, sous forme de diagrammes de séquences et d’autres scénarios. L’intervention de spécialistes en IHM permet de typer les interactions à envisager pour l’application, à partir des prescriptions générées lors de la phase précédente.

La spécification organisationnelle et interactionnelle des besoins

Cette phase détermine « qui fera quoi et quand » avec le système à concevoir. Le spécialiste « Génie Logicel » (GL) identifie pour chaque utilisateur « des cas d’utilisation à partir des descriptions précédentes des processus métiers, puis raffine les concepts métiers en composants fonctionnels appelés « objets métiers » ». La collaboration entre spécialistes du GL et de l’IHM se formalise ensuite par la transposition de ces objets métiers en objet interactionnels. Il s’agit ici de définir ce qui va représenter un concept métier dans l’interface. Cette étape inclut (Dupuy-Chessa 2011) :

- des scénarios d’interaction abstraits puis concrets, en langage naturel,
- des modèles de tâches sous forme de diagrammes dits « dynamiques » (tel que CTT, cf. § 1.10.3)
- des diagrammes dits « statiques » comme le modèle ASUR (Dubois et al. 2002) qui décrit le contexte matériel (de manière générale) et les données à traiter (illustré Figure 70),
- et des maquettes.




.

*Figure 70. Description générale du contexte matériel et des données à traiter avec le formalisme ASUR*

Si l'on reprend l'exemple illustré Figure 69, l'objet métier dégât (damage) est représenté par une « cible rouge » qui est l'objet interactionnel correspondant

L'analyse des spécifications

Il s'agit ici d'assurer la correspondance entre les objets métiers et les objets interactionnels (Figure 71). Pour cela ces objets sont structurés de manière identique en objets processus (« business process » et « interactionnal process ») et objets entités (« business entity » et « interactionnal entity »)[26]. Ensuite, on définit la correspondance (« translation ») entre les deux types d'objets. On parlera d'analyse dynamique pour ce qui est de la description des processus et d'analyse statique pour ce qui est des entités.

En ce qui concerne l'espace métier, les cas d'utilisation identifiés par le spécialiste GL sont raffinés en diagrammes de séquences pour identifier les processus (analyse dynamique). Les entités sont ensuite décrites en termes de méthodes et attributs (analyse statique).

Pour ce qui est de l'analyse dynamique de l'espace interactionnel, la spécification est complétée par des « diagrammes d'états qui décrivent le cycle de vie des objets ». Un exemple de passage d'état pour l'objet dégât est le passage de déverrouillé (éditable) à verrouillé (non éditable). L'analyse statique se fait de la même façon que pour l'espace métier.

Enfin, les objets « translation » servent, comme leur nom l'indique, à gérer la correspondance entre les objets métiers et les objets interactionnels (ils transforment par exemple les coordonnées en pixels d'un repère (objet interactionnel) en la position d'un dégât (objet métier) dans un modèle 3D). Un diagramme de séquence illustre ce processus de transformation.

[26] Un troisième élément compose les objets dits « tripartites », il s'agit de la donnée de référence servant à la nomenclature pour ce qui est de l'objet métier et représentant les données de base (comme les codes couleurs) pour les objets interactionnels.

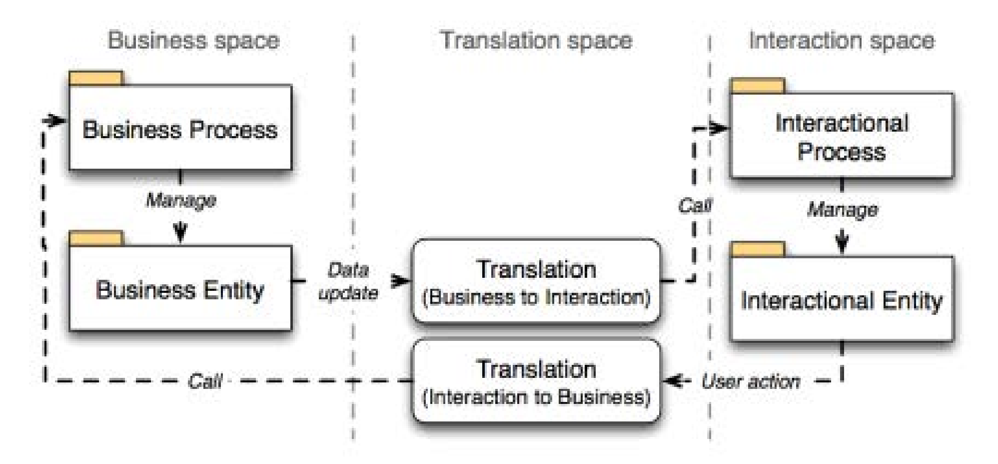



*Figure 71. L'architecture Symphony (tiré de (S Dupuy-Chessa 2011))*

### *1.20.3 La branche technique et la conception*

La branche technique consiste à décrire à la fois l'architecture logicielle et l'architecture technique.

L'architecture logicielle préconisée est le modèle MVC (cf. § 1.9.3, Figure 26). Il permet la correspondance avec les objets Symphony décrits précédemment :
- les objets interactionnels correspondent à la Vue
- les objets métiers correspondent au Modèle
- les objets translation correspondent au Contrôleur

L'analyse technique est quant à elle composée de la spécification des dispositifs choisis plus tôt (cf. modèle ASUR, (Dubois et al. 2002)), et de la description de ces dispositifs (ex. le modèle du dispositif, la résolution de l'écran, la qualité d'image...).

À partir de là, la combinaison de l'analyse fonctionnelle avec les choix techniques et le déploiement dans l'infrastructure technique peut être effectuée. Ici un nouvel élément sert de charnière, à savoir le rôle. Toujours d'après (Sophie Dupuy-Chessa et al. 2010), ce rôle commande à l'objet translation (contrôleur) la création d'un nouvel objet métier (modèle) et son instanciation en tant qu'objet interactionnel (vue). C'est ce qui se passe techniquement lors de l'identification d'un nouveau dégât qui est alors créé dans le modèle et représenté par le repère correspondant dans la vue.

### *1.20.4 Analyse critique et conclusion*

L'approche de l'équipe de Sophie Dupuy-Chessa (Dupuy-Chessa 2011) et les travaux associés aboutissent à la proposition d'une méthode de conception basée sur la mise en commun des préoccupations de l'IHM et des SI pour la conception logicielle. Plusieurs points attirent notre attention au regard de nos préoccupations.

Dès la phase d'analyse préliminaire **le métier est décrit « tel qu'il est pratiqué ».** Nous pensons qu'il y a un réel enjeu dans cette phase, notamment concernant la description d'un projet AIC. Nous cherchons en effet à tenir compte des variations d'un contexte de projet à un autre, variations qui pourront se retranscrire dans la façon de « pratiquer un métier ». Le formalisme

utilisé est le scénario, en langage naturel. L'édition est libre, sans structure qui soit déterminée par le contexte métier analysé. **Nous pensons qu'assister cette analyse par l'utilisation de concepts métiers clairement définis (tels que nous les avons introduits dans le premier chapitre) et un formalisme adéquat puissent contribuer à l'améliorer**.

Dans un deuxième temps, nous proposons d'exploiter la correspondance entre les modèles pour guider la modélisation, notamment au cours des deux phases de spécification des besoins de la branche fonctionnelle (spécification conceptuelle des besoins, spécification organisationnelle et interactionnelle des besoins). Ce lien n'est en effet pas explicite dans la méthode Symphony. Typiquement, si l'on reprend la description faite précédemment de la spécification organisationnelle et interactionnelle, il est question par exemple de définir comment « le spécialiste GL identifie des cas d'utilisation à partir des descriptions précédentes des processus métiers, puis raffine les concepts métiers en composants fonctionnels ». Cette correspondance est évoquée mais n'est pas détaillée. L'enjeu est d'utiliser les modèles adéquats, adaptés aux besoins de représentation correspondants au point de vue de chaque spécialiste.

**Nous pensons que la collaboration des différents spécialistes mise en avant dans la méthode Symphony est un point important à considérer**. La Figure 72 illustre cette collaboration au cours de la phase « spécification organisationnelle et interactionnelle. Elle débute par une activité commune de décomposition des processus métiers pendant laquelle les acteurs décrivent l'état de la situation aussi bien d'un point de vue métier que technique. Puis les spécialistes de chaque domaine modélisent leurs préoccupations ensemble ou séparément (les besoins fonctionnels par le spécialiste GL et les besoins en termes d'interaction par le spécialiste IHM et l'ergonome). L'enjeu est de permettre à chacun de formaliser son analyse puis de les lier dans un processus de conception cohérent. Cela pourra pallier au risque précédemment analysé dans (Delotte 2006), de ne pas faire la part des choses entre le travail des acteurs d'un domaine et le futur dispositif technologique qui va médiatiser ce travail. Ici la spécification organisationnelle et interactionnelle est menée parallèlement à une spécification technique pour ensuite, lors de la conception, aboutir à une ou plusieurs solutions adaptées.

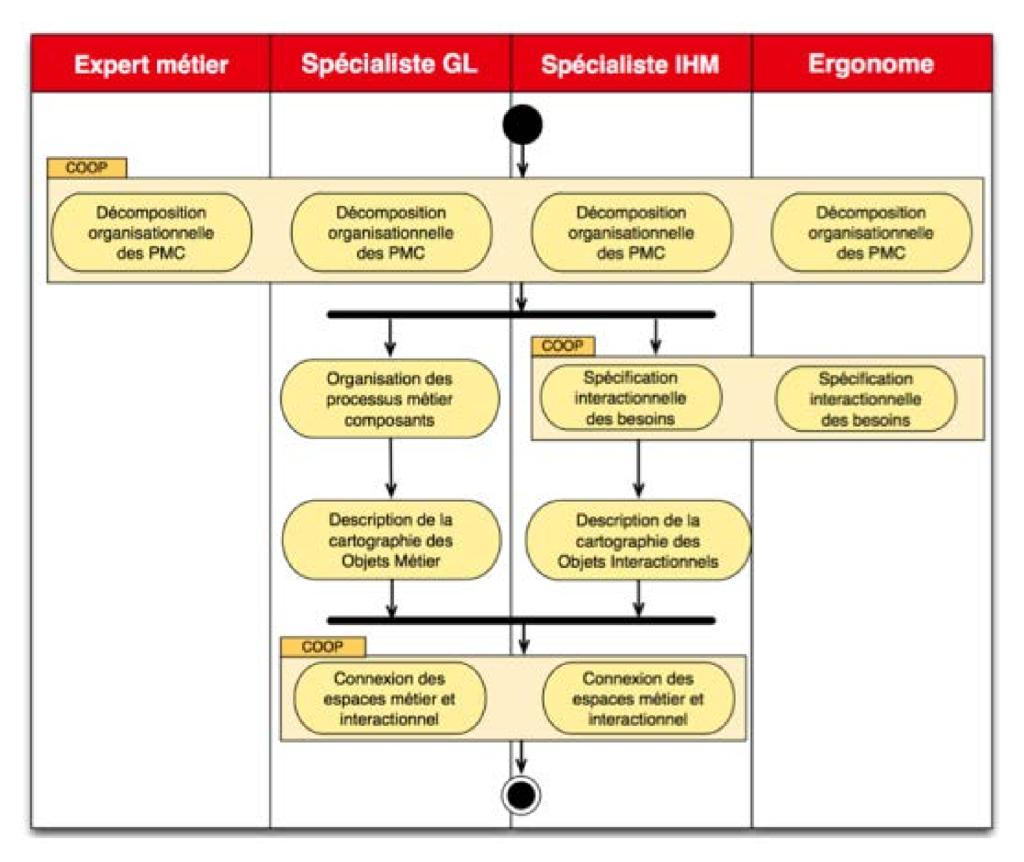



*Figure 72. Intervention des différents acteurs de la méthode Symphony au cours de la spécification des besoins (issu de (S Dupuy-Chessa 2011))*

(Dupuy-Chessa 2011) évoque l'évolution de l'espace métier déclenchée par l'intégration des choix d'interaction adoptés par l'ergonome et le spécialiste IHM. Nos contributions à des projets appliqués de développement de services pour le secteur AIC nous permettent d'argumenter dans ce sens. En effet, l'implication des acteurs métiers dans un processus de développement d'un nouvel outil passe fortement par la confrontation à des prototypages d'interfaces. Cette démarche favorise probablement très tôt l'appropriation de la technologie par ses futurs utilisateurs. **Nous pensons que l'interaction peut avoir un rôle majeur dans la conception de services.**

La troisième et dernière approche considérée dans ce chapitre sur les méthodes de conceptions concerne la conception de services spécifiques au secteur AIC, basée sur l'échange d'information et l'utilisation de maquettes numériques BIM..

## 1.21 L'IDM (Information Delivery Manual) pour la conception de services BIM

### *1.21.1 BIM, IFC, IDM : une brève introduction*

La modélisation des informations du bâtiment, plus communément « maquette numérique » ou BIM (Building Information Modeling) a pour but d'agréger les données relatives à un bâtiment en un modèle unique orienté objets (les ouvrages du bâtiment (Ibrahim & Krawczyk 2003)). Le BIM permet ainsi d'améliorer la compréhension du bâtiment (nature des ouvrages, quantité de matériaux…) et des processus mis en œuvre. Les outils associés supportent les échanges d'information dans les activités de conception et construction architecturale. Le modèle IFC

(Industry Foundation Classes, récemment renommé Information For Construction) est un format de données standardisé et ouvert, développé pour favoriser l'interopérabilité dans les approches BIM. Chaque objet IFC propose une spécification de l'information relative au projet considéré, « tout au long du cycle de vie du projet, à travers les disciplines et les applications logicielles » (Eastman & Sacks 2010). Les problématiques actuelles dans l'implémentation du format IFC sont proches de celles qui se posent pour la conception de services et SI, à savoir la maîtrise de la sécurité des échanges, la fiabilité de l'utilisation...

Le format IFC est surtout utilisé aujourd'hui pour rendre interopérables des logiciels de CAO (Conception Assistée par Ordinateur), c'est-à-dire permettre l'échange de maquettes numériques d'un logiciel à l'autre. Cependant, ces échanges basés sur l'interopérabilité des données ne sont pas structurés. Comme on peut également le lire dans (Eastman & Sacks 2010), « le schéma IFC ne capture pas la manière dont l'information est créée et partagée par les professionnels ». Il ne définit pas non plus les exigences relatives à ces échanges, ce qui rend difficile la conception de véritables logiciels de collaboration basée sur l'IFC.

Pour répondre à cela, l'approche IDM soutenue par l'association BuildingSMART a pour objectif de définir des modèles d'échanges. La réalisation de manuels IDM (Information Delivery Manual) « définit les spécifications pour faire correspondre les échanges d'information avec les objets du modèle IFC pour l'implémentation d'interfaces logicielles » réellement collaborative, dans le sens où elles supporteraient des processus métiers valides. Ce travail d'implémentation implique deux exigences primordiales :
- cibler des cas d'utilisations concrets et intéressants pour les utilisateurs,
- maintenir la compatibilité avec l'implémentation d'autres logiciels.

A l'heure actuelle, les véritables utilisateurs et concepteurs de ces manuels sont des chercheurs. Une volonté de standardisation des processus de conception/construction existe, mais la portée de l'approche permet aussi d'envisager ce formalisme pour décrire le processus collaboratif d'un projet donné.

### *1.21.2 Définition des objets d'échange sur la base de processus*

L'IDM est comme son nom l'indique un manuel, c'est-à-dire un document qui regroupe des informations nécessaires à l'implémentation de modèles IFC. Mais c'est également, selon la définition de (Berard & Karlshoej 2012), un langage de modélisation des processus métier qui étend BPMN, une manière de modéliser et reconcevoir le processus lui-même.

« Les MVD (Model View Definition) traduisent l'IDM en un document pour le développement logiciel » (Hietanen 2006). Il s'agit de sous-ensembles du modèle IFC global, et contenant la structure des données pour l'échange.

<u>Le processus et les exigences d'échange</u>

Le processus illustré dans la figure suivante (Figure 73) représente la collaboration entre plusieurs rôles organisationnels que portent les *concepteurs* (design team), le *coordinateur* (project management) et le *fournisseur* (supply chain). On peut y lire que l'équipe de conception

conçoit son bâtiment en utilisant des produits qui sont développés par les fabricants qui valident et estiment le prix de ces derniers. Le coordinateur doit assister ce processus.

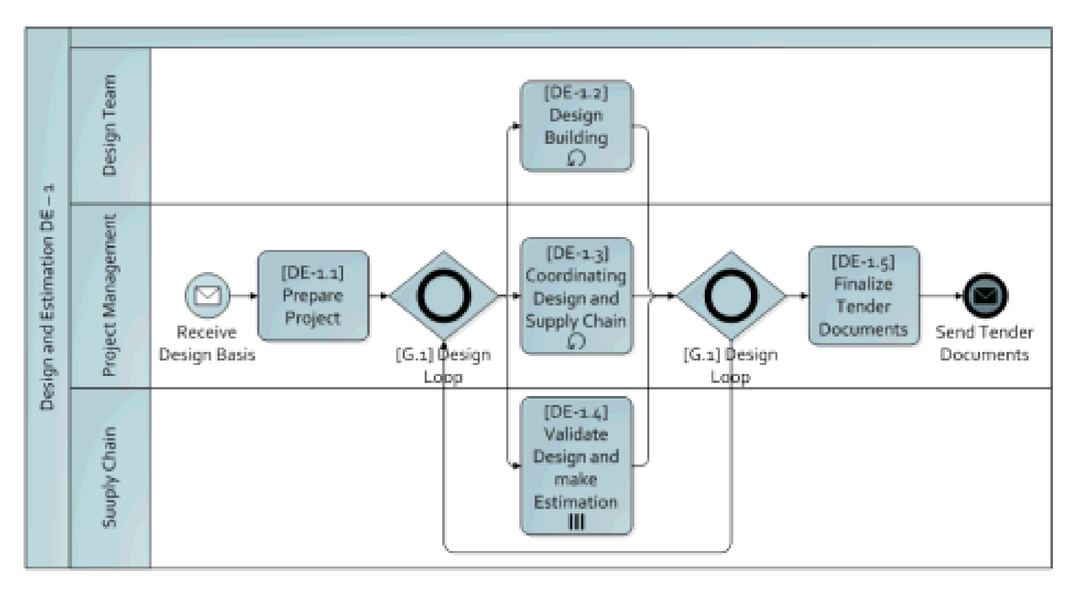



*Figure 73. Vue d'ensemble d'un processus (Berard & Karlshoej 2012)*

Dans le cadre de l'utilisation d'un service « d'ouvrages virtuels » (Virtual Building Product, VBP) à travers le BIM, les actions métiers identifiées dans le processus sont décomposées pour former un nouveau processus qui décrit le service utilisé ((Charles Eastman & Sacks 2010) utilise le terme « cas d'utilisation »). Dans l'exemple décrit ci-dessous (Figure 73), l'action « Design Building » est décomposée en cinq actions : « load VBP », « Choose size and type », « Validate ease of manufacturing », « Insert into model » et « Update design model » (Figure 74). Notons que l'action « validate ease of manufacturing » se fait automatiquement (l'acteur responsable est le service « virtual building product» lui-même) en fonction de règles prédéfinies.

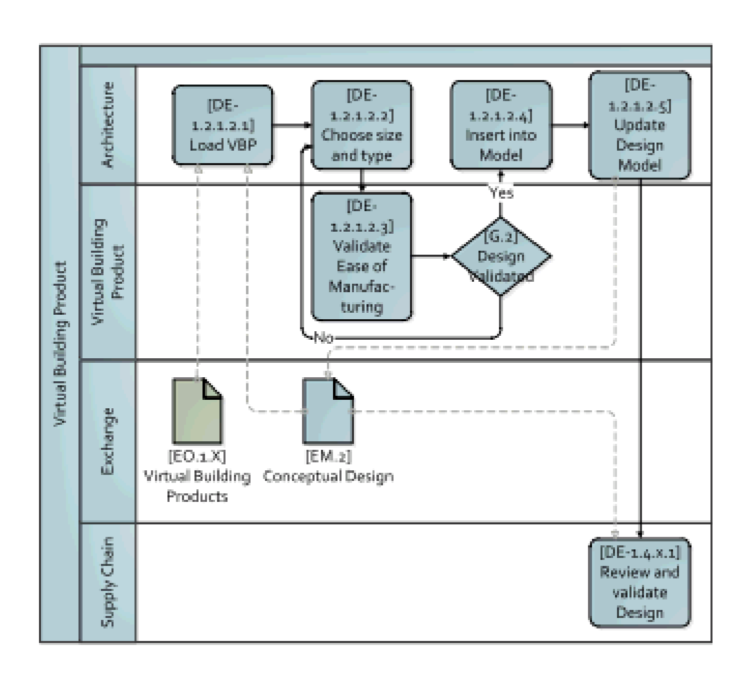



*Figure 74. « Cas d'utilisation » du produit virtuel (Berard & Karlshoej 2012)*

L'enjeu est alors de pouvoir définir les exigences d'échange en identifiant les « objets d'échange » (E.O) et les «modèles d'échange » (E.M) qu'ils constituent. Les E.O évoluent en fonction du modèle auquel ils appartiennent. Par exemple, un poteau pourra être un simple volume géométrique dans un modèle conceptuel ou être plus finement décrit lors des phases ultérieures (Eastman & Jeong 2010). Ces objets sont soumis à des règles métiers comme par exemple des dimensions réglementaires.

L'implémentation

La création de l'IDM autour de la modélisation des processus et des objets d'échange permet de formaliser les besoins des professionnels tels qu'ils les communiquent. C'est une modélisation, indépendante du format utilisé, car les professionnels n'ont généralement pas les compétences nécessaires pour définir cela. En revanche, « la correspondance de l'ensemble d'informations avec une structure de données est la responsabilité du développeur du système » (Berard & Karlshoej 2012). Comme le montre le schéma suivant (Figure 75), il s'agit d'appliquer le schéma IFC et sa documentation (« IFC model specification »[27]) aux exigences d'échange pour implémenter une solution. Les « Model View Definitions » (MVD), qui sont des sous-ensembles de modèle IFC, documentent la manière dont ces spécifications sont appliquées pour l'implémentation d'une solution IFC (Hietanen 2006). Enfin, celle-ci est implémentée.

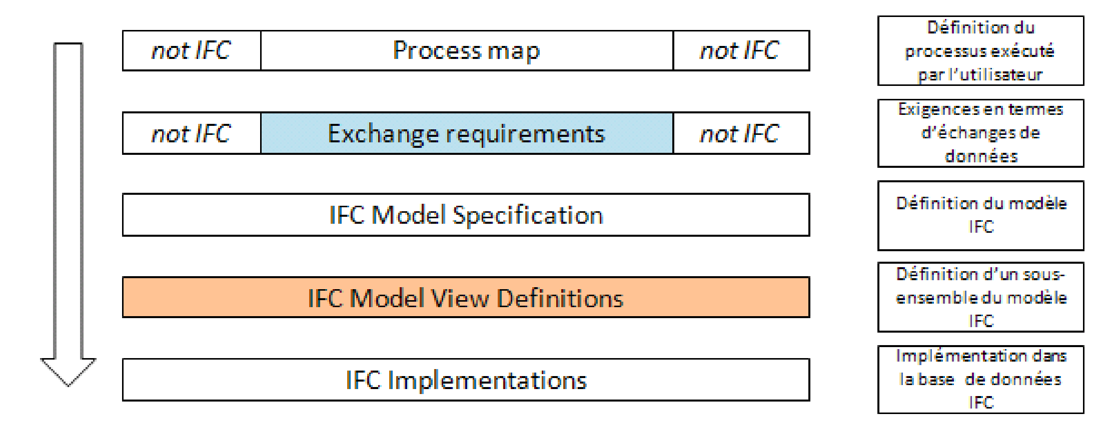



*Figure 75. Vue d'ensemble des étapes pour le déploiement de solutions basées sur l'IFC (adapté de (Hietanen 2006))*

Le modèle IFC est aujourd'hui entièrement certifié ISO, ce qui va encourager sa standardisation et son adoption par le secteur. La portée de cette approche est de pouvoir écrire des processus métiers autour du BIM de manière standardisée également afin d'en créer des catalogues. A termes, cela permettra d'implémenter des services BIM par la composition de processus standards, liés aux besoins des utilisateurs.

### *1.21.3 Analyse critique et conclusion*

Le BIM et la centralisation des données du bâtiment autour d'un modèle unique est un contexte technologique innovant et plein d'intérêt pour le secteur AIC. Il est supporté par la création de données dans un format d'échange standard : l'IFC.

[27] on pourrait d'ailleurs parler de méta-modèle IFC

Cependant, comme l'évoque (Eastman & Sacks 2010), « le schéma IFC ne capture pas la manière dont l'information est créée et partagée par les professionnels ». Pour cela il est nécessaire d'interroger les professionnels et de capter les processus qu'ils mettent en œuvre dans leur contexte de travail spécifique. L'IDM est conçu à cet effet : il rassemble les processus et les exigences en termes d'échanges en dehors de toute préoccupation sur l'implémentation.

Les processus de l'IDM sont créés en plusieurs temps. Dans un premier temps, c'est le point de vue métier qui est modélisé (Figure 73), avec l'analyse des processus collaboratifs mis en place au cours de phase spécifique du projet. Basé sur le BPMN, ces processus décrivent les tâches de chacun sous forme de « swimlanes ». Le deuxième point de vue porte sur des « cas d'utilisation » au cours desquels les « tâches métiers » sont décomposées en « tâches utilisateur » sur le système (Figure 74). Le formalisme utilisé est le même et la décomposition des tâches est identifiable par le nommage de celles-ci (ex. les tâches 1.2.1 et 1.2.2 sont les tâches utilisateurs qui composent la tâche métier 1.2). L'étude ne nous dit cependant pas qui exprime ces points de vue et comment. Il serait intéressant d'identifier la place des utilisateurs, des analystes métiers, ou encore des développeurs dans un tel processus ainsi que la manière dont ils collaborent autour des langages utilisés.

Concernant le formalisme, on ne retrouve par contre pas de formalismes propres au génie logiciel (GL). Par exemple, les auteurs utilisent le terme « cas d'utilisation » qui est emprunté au GL mais qui n'est pas formalisé par le diagramme UML correspondant (le diagramme de cas d'utilisation).

Vis-à-vis de l'analyse métier, nous pensons que les modèles de processus sont inadaptés pour la description du contexte particulier d'un projet AIC (cf. § 1.14.4). (Berard & Karlshoej 2012) partage notre avis, argumentant en ces termes : « les processus de projet de construction sont très flexibles et, avec les pratiques d'aujourd'hui, il est difficile de modéliser un processus très détaillé afin de le standardiser ». Il ajoute : « non seulement l'ordre de l'exécution des tâches est différent d'un projet à un autre mais l'interaction entre organisations peut aussi différer à l'intérieur même d'un projet ». L'objectif est donc d'éviter d'investir du temps en se perdant dans de la « surmodélisation » et de trouver l'équilibre entre modélisations générique et spécifique afin de couvrir différents besoins tout en restant pertinent. Nous pensons qu'il est important de privilégier des modèles simples autour de concepts clairement définis.

Dans le cadre du développement d'un ensemble complet de services basés sur le BIM, l'IDM décrit non pas un seul, mais de multiples processus qui composent l'activité globale de projet. Au final, un IDM se présente comme une « checklist de plus de 50 pages ». Nous pensons qu'il serait utile de décomposer cette approche en créant des IDM plus « ciblés ». Il s'agit de mener des processus de développement plus courts, afin d'en faciliter le déroulement et la traçabilité.

Nous remarquons dans (Eastman & Sacks 2010) que les processus de l'IDM modélisent également l'information qui n'est pas échangée à travers des modèles. Cela permettrait d'étendre le champ d'application en dehors du BIM et nous y voyons un intérêt pour la spécification de services IT plus variés, relatifs à d'autres technologies (collecticiels, applications mobiles…). L'approche IDM ne s'étend pas sur la proposition de services autres que le BIM, pourtant elle pourrait y contribuer. En effet, elle fournit une analyse métier cohérente et complète qui pourrait être intégrée aux méthodes plus générales de Génie Logiciel et de conception de services. Cela

serait notamment profitable aux professionnels du secteur à court terme, en attendant la démocratisation du BIM et des IFC qui se fait lentement.

## 1.22 Conclusion et mise en place de la méthode

Le but de cette partie était d'introduire la conception de services collaboratifs pour comprendre la démarche à suivre dans le développement de notre propre méthode de conception. Notre objectif est de répondre aux problématiques sectorielles détaillées au chapitre 3 (notamment au § 1.7.2). Ce chapitre détaille trois méthodes de conception dont nous avons pu faire ressortir les avantages et limites. Il complète notre analyse menée au travers des chapitres 4, 5 et 6 sur le GL, les IHM, les SI et le TCAO.

De manière générale, ces méthodes nous permettent de comprendre le déroulement d'un processus de conception dans un cas concret d'application. Elles ne semblent cependant pas faire leurs preuves dans des cas réels de projet de conception et transfert des services, à savoir dans le but de répondre aux besoins spécifiques d'un domaine professionnel et dans des conditions réelles de développement. De plus, chacune couvre un contexte technologique très précis, ce qui limite leur extension dans des domaines variés.

Pour approfondir certains des points d'intérêts évoqués ou pallier à certaines des limites perçues, nous proposons de définir le cadre de notre approche sur la base des hypothèses suivantes :

- Une analyse métier doit être basée sur la modélisation de situations collaboratives précises et non sur la « surmodélisation » de processus complexes.
- Les modèles graphiques favorisent la synthèse et la compréhension, ils sont particulièrement adaptés à la mise en correspondance de concepts entre plusieurs modèles. Les modélisations littérales fournissent quant à elles de l'information précise et en grande quantité. Il sera judicieux de combiner les deux en fonction de l'information à modéliser.
- Le couplage tardif entre exigences et aspects techniques assure la possibilité d'étendre l'approche à différents domaines métiers comme à différents contextes technologiques.
- Se concentrer sur la notion d'information manipulée (à travers le concept d'objet) permet de traduire l'espace organisationnel en espace interactionnel, avant de définir les services d'un point de vue fonctionnel.
- Plusieurs espaces interactionnels et fonctionnels peuvent correspondre à plusieurs espaces organisationnels, en fonction des variations de contexte, que ce soit celui de l'activité, de l'acteur ou de l'utilisateur.

A partir des différents concepts qui émergent de nos analyses, nous proposons une approche structurée basée sur un cheminement d'étapes relatives à trois points de vue (Figure 76) :

- Le **point de vue organisationnel**. Nous souhaitons être plus précis que lors de l'énoncé de besoins ou de principes généraux relatifs à la collaboration mais sans atteindre la rigidité des processus. Nous nous concentrons sur l'analyse des rôles et activités métiers des utilisateurs dans une équipe de projet AIC. Nous adoptons le concept de **Pratiques métiers (Collectives et Individuelles)**.
- Le **point de vue opérationnel**. Nous considérons ici les personnes en tant qu'utilisateurs de diverses technologies et dans des contextes d'utilisation relatifs au projet AIC pour concevoir un scénario d'utilisation adapté (Il nous semble important de n'avoir que rarement recours à l'utilisateur). Nous adoptons le concept d'**Usage**.
- Le **point de vue fonctionnel**. Notre objectif est de spécifier des nouveaux services ou d'adapter des services existants pour améliorer l'utilité et l'utilisabilité des outils de TCAO. Nous adoptons le concept de **Services Collaboratifs**.

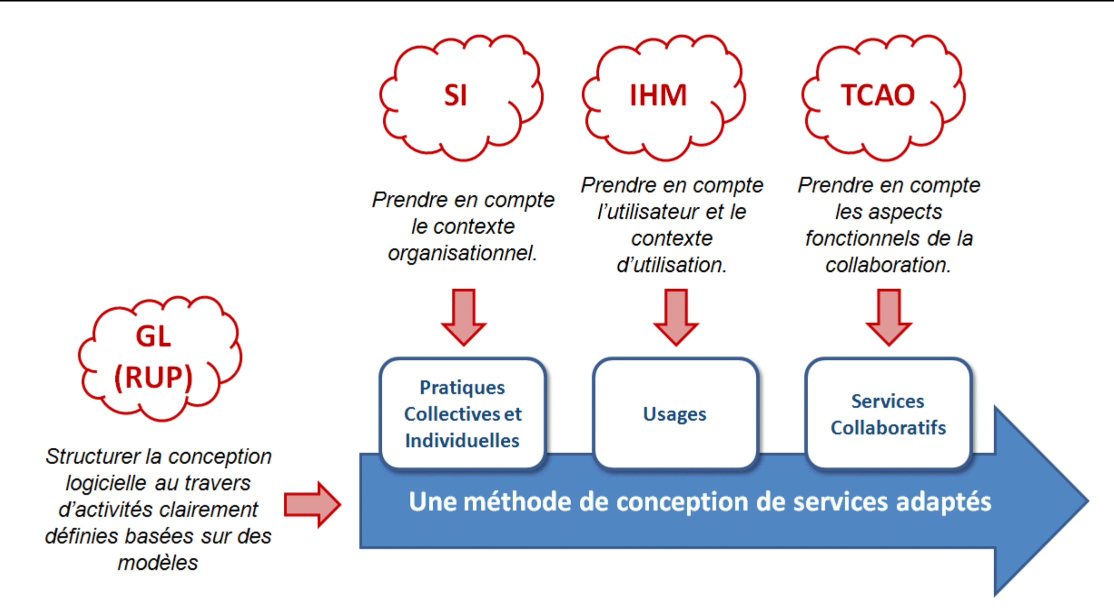



*Figure 76. Construction de notre méthode en fonctions des approches analysées et leurs apports respectifs*

# PARTIE 3 : Guider la conception de services collaboratifs adaptés au secteur de la construction. Études, propositions et perspectives.

Nous montrons dans cette dernière partie la construction de notre méthode d'analyse et conception pour la proposition de services collaboratifs adaptés sur la base des hypothèses dégagées et de l'approche introduite. Le premier chapitre décrit la méthodologie adoptée, et les suivants présentent chaque étape de l'approche autour de la méta-modélisation et modélisation des pratiques, des usages et des services. Rappelons qu'une approche qui s'inscrit dans une démarche scientifique de conception se distingue d'une approche d'ingénierie classique car elle mêle propositions et évaluations dans un processus continu, depuis la première idée jusqu'à la réalisation. Nous accordons autant d'importance à la description des concepts introduits qu'aux cas d'études qui ont permis de les identifier et les valider. Un dernier chapitre fait état de ces cas d'études, nous permettant de tirer des conclusions sur la pertinence et l'applicabilité de cette méthode.

# Chapitre 8 – Introduction de la proposition

Ce chapitre dégage dans un premier temps les enjeux et objectifs de cette étude. Ils sont issus des contextes de développement de notre méthode de conception, à savoir le contexte d'application (le domaine AIC) et le contexte d'étude (créé par les deux laboratoires encadrant cette étude, le MAP-CRAI et le CRP Henri Tudor). Il présente ensuite la méthodologie adoptée, les outils utilisés et le processus mis en œuvre pour déployer une méthode de conception favorisant les opportunités métier et technologique : la méthode que nous nommons PUSH pour « Practice and Usage-based Service enHancement ».

## 1.23 Enjeux de la méthode

### *1.23.1 Un domaine d'application particulier*

Notre problématique d'adaptation des services informatiques est issue d'un constat initial que nous avons introduit dans les premiers chapitres de cet ouvrage : un domaine particulier implique des services particuliers, les services génériques ne répondant pas totalement aux besoins des professionnels. Un domaine collaboratif (caractérisé par le travail coordonné de plusieurs acteurs) nécessite des services de support à la coproduction, à la communication et la coordination dans un contexte de travail souvent distant et réparti dans le temps (cf. 1.16.1).

Pour étudier comment la spécificité d'un domaine peut influencer le développement de services nous nous sommes concentrés sur le domaine AIC. Rappelons que le projet de conception et construction architecturales diffère d'un processus industriel notamment car l'objet produit, c'est-à-dire le bâtiment, est unique. Il est également soumis à des garanties décennales qui impliquent la responsabilité des concepteurs quant à la qualité de l'ouvrage dans le temps. Les tâches « métier » en conception et construction sont non répétitives et sans cesse ajustées de par le contexte de projet. Ce contexte, appelé « contexte de l'activité collective » dans les travaux antérieurs (voir 1.1), évolue d'un projet à un autre mais aussi au cours même du projet sous des « influences extérieures » pareilles à celles qui opèrent sur le « business model » des entreprises (voir 1.13.4) : l'environnement,

l'économie, la technique... Avec lui varient les autres contextes liés à l'usage des outils pour assister le projet : le contexte utilisateur (physique et matériel) et le contexte de l'acteur (cognitif) (cf. § 1.2.3).

Aujourd'hui, les solutions logicielles pour le partage d'information sont nombreuses. Elles existent par exemple sous la forme pièces jointes dans un mail (ou via un message instantané) ou de stockage dans le « cloud » (GoogleDrive, Microsoft SkyDrive, DropBox…). Cependant, dans le contexte professionnel d'un projet de construction, ces outils trouvent rapidement leurs limites (malgré des versions dédiées entreprises). Le manque d'espace disponible, d'archivage, de sécurité et de traçabilité ou encore les règles de confidentialité inadaptées limitent leur adoption par les professionnels qui privilégient pour certains des solutions « privées » développées en interne… Les outils tels que CRTI-weB et plus particulièrement son service « documents » (cf. 1.4.3), se démarquent par des fonctionnalités pensées pour les activités et par les acteurs du domaine de la construction, telles que l'utilisation d'une convention de nommage « métier », le filtrage par émetteur de plans, les commentaires, la validation… Ils présentent aussi d'autres qualités comme la sécurité et la confidentialité.

### *1.23.2 Le contexte d'étude et les objectifs*

Le double contexte offert par le CRAI (Centre de Recherche en Architecture et Ingénierie) de Nancy et le CRP (Centre de Recherche Public) Henri Tudor à Luxembourg, a permis d'inscrire des objectifs de recherche dans la continuité de travaux théoriques et dans des cas d'études réels et pertinents. La thématique de recherche du CRAI est la proposition de méthodes et outils pour l'assistance à la conception, la construction et la coordination dans le domaine AIC. C'est avant tout un contexte académique. Les objectifs du CRP Henri Tudor sont quant à eux relatifs à la recherche et au développement de services innovants pour le transfert vers les professionnels. Un des programmes de l'organisation est dédié au domaine de la construction. Par rapport à la structure de notre recherche introduite au § 1.8.2 (et comme le rappelle la Figure 77) ce contexte a donc alimenté à la fois notre base de connaissances (en jaune sur la figure, cf. Partie 1) et notre terrain d'expérimentation (en bleu, présenté au Chapitre 12 de ce manuscrit). Il a permis d'allier préoccupations métiers et opportunités d'innovation.

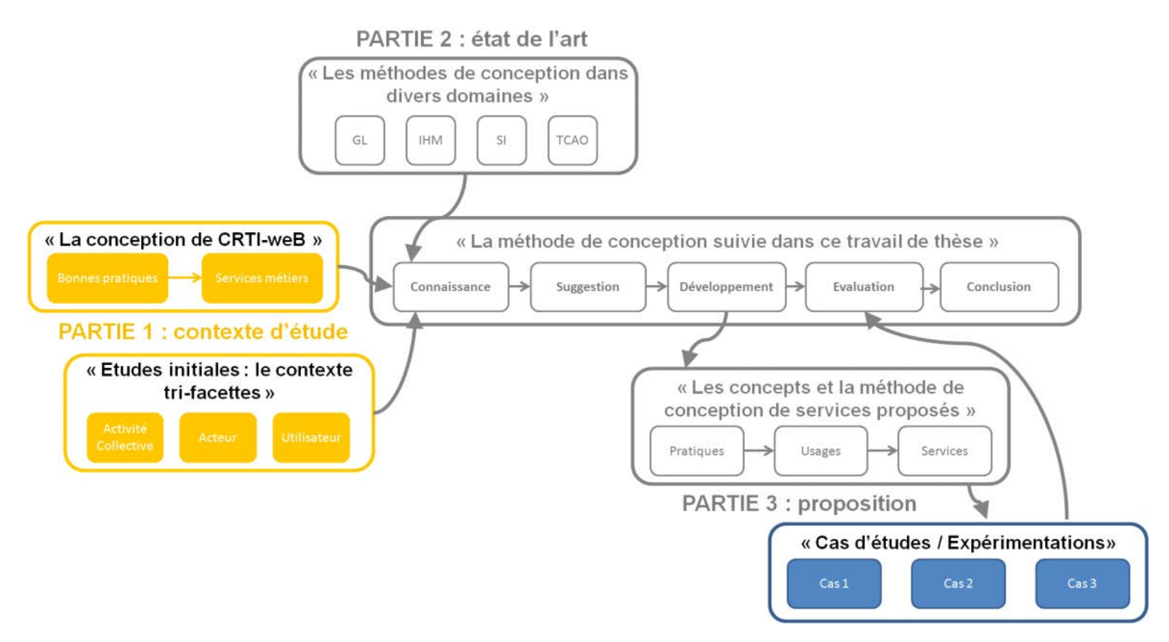



*Figure 77. Parties de la recherche supportées par le contexte d'étude*

Les besoins d'adaptation des services pour le domaine AIC et par conséquent le besoin d'une méthode de conception ont été identifiés dans la littérature et par l'observation de cas concrets d'utilisation de services, notamment ceux de l'outil CRTI-weB (cf. § 1.5). En remettant en cause les services qui paraissent peu adaptés de par les retours des utilisateurs, nous nous sommes servis de CRTI-weB et de l'analyse de projets de construction hébergés sur cette plateforme comme point de départ de notre étude. À la lumière des théories analysées au cours de notre état de l'art sur les méthodes de conception logicielle, nous avons déployé notre propre méthode en l'appliquant dans ce contexte spécifique.

Les développeurs sont actuellement responsables des ateliers de formation avec les utilisateurs de CRTI-weB et recueillent également leurs retours réguliers. Au cours d'échanges avec eux, nous avons relevé deux enjeux majeurs :

- faciliter le dialogue et éviter les incompréhensions, afin de lever le risque de développer des services couteux mais s'avérant inadaptés,
- garder en mémoire les décisions prises afin de tracer l'évolution du projet développement.

La section suivante montre comment la caractérisation des trois points de vue (organisationnel, opérationnel et fonctionnel) définie en conclusion du chapitre précédent permet de répondre à ces enjeux.

## 1.24 Méthodologie

### *1.24.1 La caractérisation des points de vue*

Notre approche de conception de services adaptés est basée sur l'expression des points de vue organisationnel, opérationnel et fonctionnel par la modélisation des pratiques métiers, des usages d'une technologie et des services. Ces modèles (M1) décrivent des cas réels sur base de concepts décrits dans trois Méta-Modèles (M2) qui sont agrégés en un seul : le Méta-Modèle des Services

Adaptés (MMSA). Les modélisations successives nous permettent de raffiner les méta-modèles proposés dans un premier temps à partir de la littérature analysée (Figure 78).

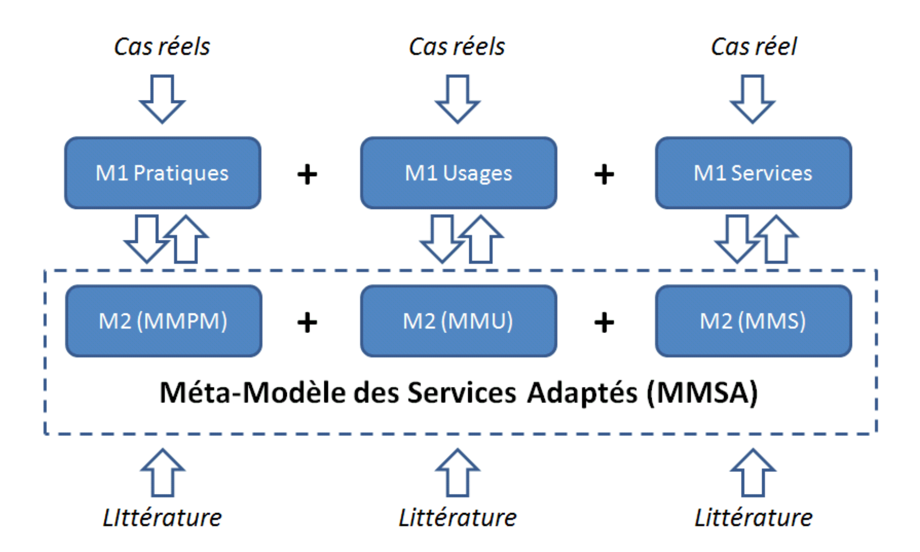



*Figure 78. Processus de méta-modélisation à partir de l'analyse de cas et de la littérature dans différents domaines*

Dans des travaux de recherche antérieurs, menés dans le cadre du projet de recherche Dest2Co par le CRP Henri Tudor, les points de vue et les modèles sont la base d'une méthode de conception de services : la méthode Dest2Co (Zignale et al. 2011). Cette méthode propose un processus qui intègre trois étapes (appelées vues, Figure 79) :

- la « business requirements view » (BRV), relative au point de vue des experts métiers, qui contient les modèles décrivant les exigences générales du domaine (sous forme de pratiques collectives),
- la « business solution view » (BSV), relative au point de vue des experts en services, qui décrit la solution en termes de services métiers, sans aucune considération technique,
- la « technical solution view » (TSV), qui correspond a une phase d'analyse et conception logicielle, assurée par les experts du génie logiciel.

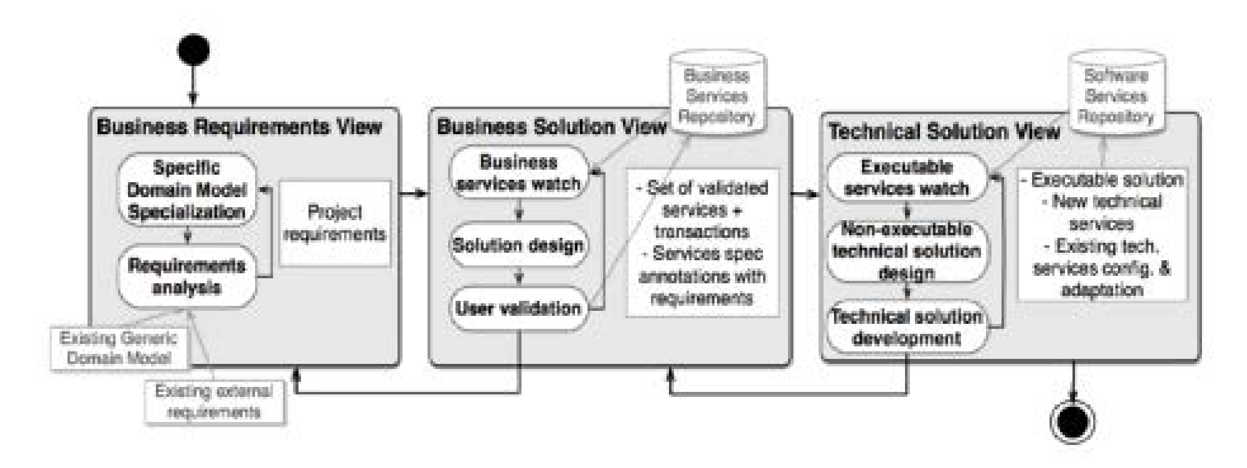



*Figure 79. Le processus de la méthode Dest2Co (tiré de (Zignale et al. 2011))*

En comparant le projet de conception de service à un projet architectural, nous pouvons également créer un parallèle avec le concept de points de vue traité dans (Hanrot 2005). On peut y lire que « le point de vue porte sur certains aspects de l'objet » (ici architectural mais dans notre cas nous parlons de service). L'auteur ajoute que le point de vue dépend « de l'expertise de l'acteur » et « de sa connaissance des autres domaines ». Cette description du concept de point de vue correspond à celle que nous lui conférons dans notre étude. Il s'agit en effet de faire intervenir des acteurs aux expertises différentes (expert métier, concepteur en génie logiciel et IHM, développeur) non pas pour évaluer mais pour assurer la qualité du service. La modélisation, activité principale dans le projet architectural, est également au cœur de notre approche de conception. Elle doit permettre d'éviter des ambiguïtés, et augmenter la compréhension que les intervenants ont du travail réalisé par les autres intervenants d'un projet de conception de services.

Notre étude s'appuie sur cette notion de point de vue en proposant également un processus composé de trois étapes.

Le **point de vue organisationnel** porte sur les pratiques des acteurs d'un projet AIC (en fonction de leur rôle), ayant des besoins spécifiques dérivés des pratiques collectives qui se mettent en place au cours du projet. De par le contexte professionnel dans lequel se trouve notre étude, nous nous sommes essentiellement intéressés aux besoins liés au partage de documents des architectes avec d'autres acteurs (dont notamment les entreprises, les ingénieurs et le maître d'ouvrage) et à la visite de chantier. L'analyse de pratiques a contribué à améliorer notre compréhension des aspects métiers relatifs à la modélisation de la collaboration. Les travaux précédents sur la méta-modélisation du contexte coopératif (chapitre 1) et notre étude sur les processus métiers (chapitre 5) nous servent de base à la création du méta-modèle des pratiques métiers (MMPM).

Le **point de vue opérationnel** prend en considération la médiatisation des pratiques par les usages d'outils dédiés. Les outils qui nous intéressent particulièrement sont les outils d'assistance à la collaboration dont notamment les plateformes d'échange de document. Les questions relatives à l'usage concernent :

- la définition du contexte de l'utilisateur,
- la définition de l'outil,
- la définition de l'interaction entre l'utilisateur et l'outil,
- la définition de l'information manipulée à travers l'outil.

Enfin d'un **point de vue fonctionnel**, il est nécessaire de tenir compte des caractéristiques techniques de l'outil utilisé tout en implémentant au mieux les services spécifiés.

L'ensemble de l'état de l'art sur les approches et méthodes de conception (Partie 2) nous permet de comprendre les concepts à manipuler pour l'analyse des usages et la spécification des services collaboratifs. Cela nous mène à la construction de deux autres méta-modèles : le Méta-Modèle d'Usages (MMU) et le Méta-Modèle de Services (MMS). Il s'agit notamment de choisir les éléments les plus pertinents afin d'avoir assez d'information sans pour autant en surcharger les destinataires des analyses (le concepteur pour l'usage et le développeur pour le service).

Par l'association des trois méta-modèles (MMPM, MMU et MMS) nous créons le Méta-Modèle de Services Adaptés (MMSA). Il s'agit de décrire comment les usages « médiatisent » les pratiques

métiers puis comment ils sont « matérialisés » par des services. C'est pourquoi les services décrits sont dits « adaptés », sous-entendu adaptés aux pratiques métiers étudiées.

### *1.24.2 Les formalismes et outils utilisés*

Nos méta-modèles sont représentés sous la forme de diagrammes de classes UML. Nous avons utilisé l'outil StarUML pour les éditer. Comme l'illustre l'exemple de la Figure 80, nous retrouvons dans ce type de diagrammes :

- des classes (famille, parents…) et les attributs qui les caractérisent (nom, âge, profession…). Les attributs peuvent être déterminés par la saisie d'un texte libre (ex. nom) ou par un choix parmi une suite de valeurs (ex. type_profession). Le choix entre oui et non (ex. marié) est une énumération particulière. Le choix d'une valeur numérique peut être spécifié par une plage de valeurs.
- Des associations entre concepts avec les cardinalités qui les définissent (ex. le père est en couple avec une seule mère et réciproquement (relation 1 - 1)), les fils sont frères de aucun ou plusieurs autres enfants (relation 0.*)…),
- des compositions (une famille est composée de parents et d'enfants) pour lesquelles nous ne définissons pas de cardinalités en considérant le cas le plus général à chaque fois (0.*)
- des généralisations (les fils et les filles sont des enfants), relations à travers lesquelles les sous-classes héritent des attributs de la classe au dessus.

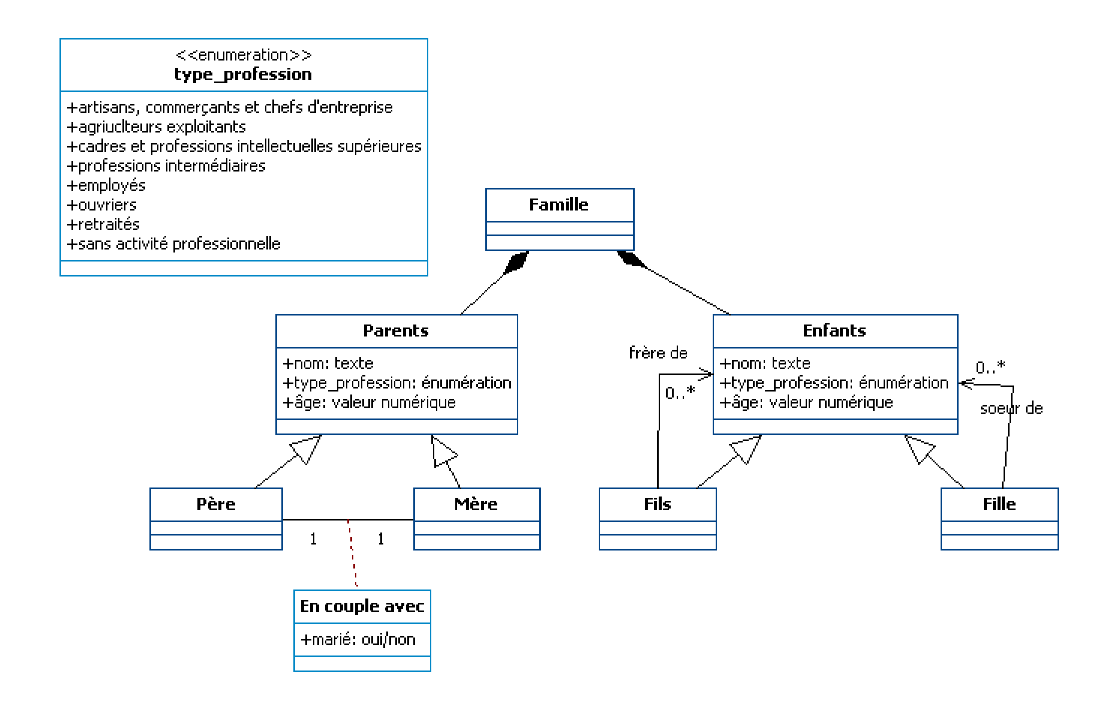



*Figure 80. Exemple de diagramme de classes UML caractérisant une famille*

Ces méta-modèles sont instanciés par des modèles dont la forme est décrite par un formalisme particulier, selon le besoin de représentation. Nous utilisons notamment les diagrammes de cas d'utilisation, de tâches ou encore de séquence (cf. § 1.9.3) qui sont des formalismes adaptés et adoptés dans les domaines de la conception logicielle. Le tableau est un formalisme simple qui permet de représenter de manière claire et structurée des données littérales (de type texte court) ou numériques. Il est ainsi suffisant pour modéliser certains concepts qui ne demandent pas une représentation

graphique particulière. Le tableau représente les données sous forme de cases alignées dans des colonnes et des lignes.

Cependant, les formalismes existants ne permettent pas d'instancier entièrement notre MMSA. Nous avons donc créé nos propres formalismes en utilisant l'environnement de développement Eclipse et plus particulièrement le framework GMF (Graphical Modelling Framework) extension d'un autre framework qu'est EMF.

Le framework GMF et l'exploitation de nos méta-modèles

Le projet Eclipse[28] est un ensemble de projets de développement logiciel autour de la plateforme Eclipse, un environnement de développement intégré principalement écrit en Java. Le projet « Eclipse Modeling » a été lancé par IBM dans le but d'unifier leurs outils de développement utilisant des modèles. Comme nous l'avons vu dans la section introduisant l'ingénierie dirigée par les modèles (cf. 1.11.1), l'espace technique EMF (Eclispe Modeling Framework) est similaire à celui de l'architecture dirigée par les modèles MDA.

Ainsi, un projet EMF permet de créer des méta-modèles « ecore » (M2) basé sur le langage EMOF (Essential Meta-Object Facility) puis les modèles (M1) conformes à ce langage. Pour représenter le méta-modèle, le framework propose le formalisme « ecorediag » proche de celui du diagramme de classes UML. Nous pouvons donc implémenter nos propres méta-modèles en EMF sans changement de formalisme. Les éléments de modèle (M1) créés à partir des concepts du M2 sont représentés sous forme d'une liste, leurs attributs étant éditables à partir d'une fenêtre « propriétés ».

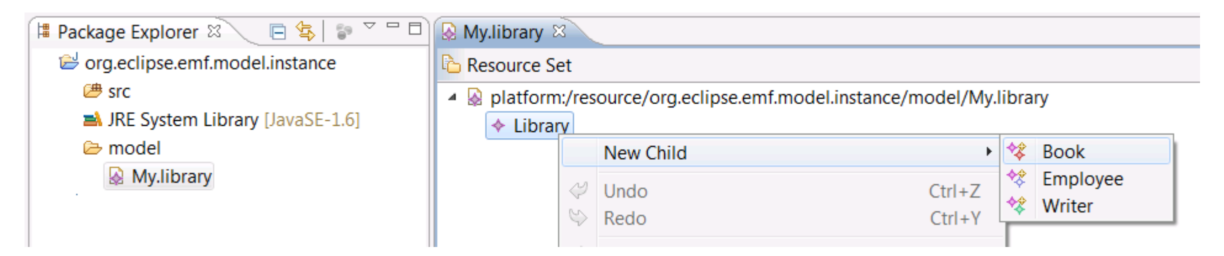



*Figure 81. Exemple d'instanciation d'une classe de méta-modèle avec l'éditeur EMF*

Le framework GMF (Graphical Modeling Framework) est né de l'association d'EMF à un autre framework : GEF (Graphical Editor Framework). Ce dernier permet la création d'éditeurs graphiques riches. Ainsi, GMF permet l'instanciation d'un méta-modèle « ecore » par des diagrammes, par la génération de ce genre d'éditeurs entièrement personnalisables.

Le processus de création et personnalisation de l'éditeur est assisté par un « dashboard » (Figure 82). Après la création du méta-modèle « ecore » et la génération des classes java, deux modèles sont créés pour décrire l'aspect graphique des diagrammes et la palette de l'éditeur : le « graphical def model » et le « tooling def model ». Dans le « tooling def model » il est par exemple possible d'agréger les outils de la palette dans des sous-menus ou de les séparer par des traits afin de mieux les

[28] http://www.eclipse.org/eclipse/index.php

organiser. Grâce au « graphical def model », nous avons exploité une série de paramétrages basiques mais qui répondent à nos besoins de personnalisation des formalismes :

- changement des formes utilisées pour représenter les éléments (rectangles, rectangles, arrondis, ellipses) et variations du trait utilisé (style, épaisseur, couleur),
- variations des traits utilisés pour les relations,
- variations des polices pour chaque attribut utilisé,
- ajout d'un titre pour chaque attribut,
- affichage d'un icône pour chaque élément…

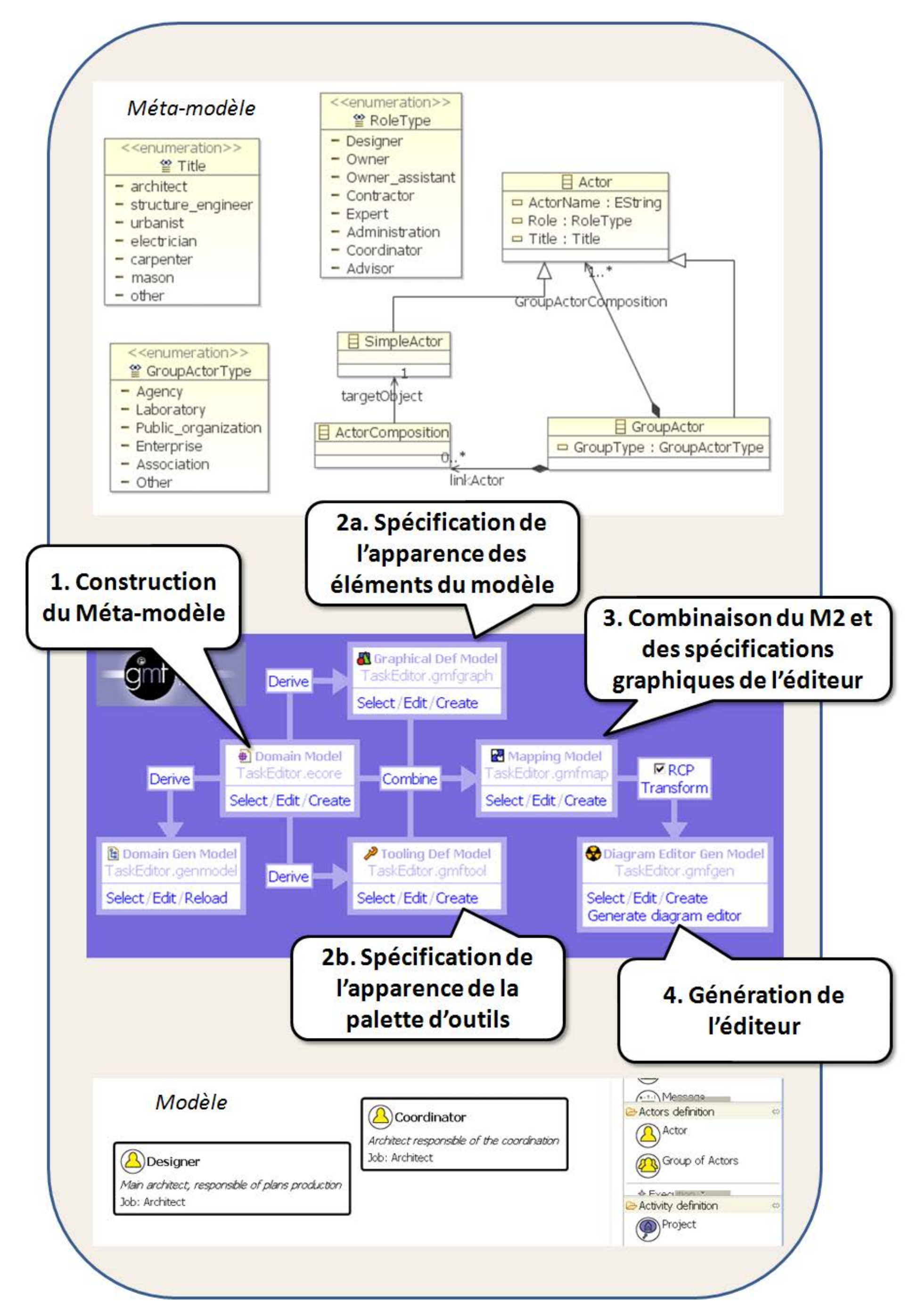



Figure 82. Processus de génération d'un éditeur de modèles avec GMF

Positionnement par rapport à l'architecture dirigée par les modèles

Le cadre de notre étude ne couvre pas les développements et déploiements automatisés des services informatiques couverts par certaines approches de MDA (architecture dirigée par les modèles). Nous nous concentrons sur la spécification de ces services et leur développement dans le cadre d'expérimentations en collaboration avec des développeurs. Aussi, nous prenons le parti de proposer des modèles contemplatifs et non productifs (Favre 2004). Nous rappelons que les modèles contemplatifs sont utiles pour la communication et la compréhension mais ne sont pas utilisés pour la production du code qui reste l'activité de l'informaticien (cf. § 1.11.1).

Ce champ d'études fait l'objet de plusieurs travaux de recherche (cf. § 1.11.2) qui le font évoluer. Dans la pratique, la génération n'est pas encore entièrement automatisée et les cas d'applications restent des cas relativement simples. Cependant, il serait intéressant d'inclure notre approche dans une telle démarche pour en explorer les enjeux : cela consisterait à automatiser le passage de la modélisation des pratiques métiers à la génération de services qui les supportent. Eclipse propose des outils de transformation de modèles comme ATL[29] qui pourraient nous permettre d'investir une recherche dans ce sens et pourrait faire l'objet d'une perspective de ce travail.

## 1.25 La méthode PUSH : « Practice and Usage-based Service enHancement

### *1.25.1 Un processus qui valorise les opportunités*

Notre approche de conception de services adaptés préconise l'expression de trois points de vue distincts mais conceptuellement liés : le point de vue organisationnel, le point de vue opérationnel et le point de vue fonctionnel. Des méta-modèles caractérisent les concepts manipulés dans ces trois points de vue et différents outils et formalismes permettent de les exprimer. Comme l'illustre la Figure 83 le processus de conception supporté par l'expression de ces points de vue devra favoriser la valorisation des opportunités aussi bien métier que technologiques. Les champs d'études analysés dans notre état de l'art illustrent cette double valorisation.

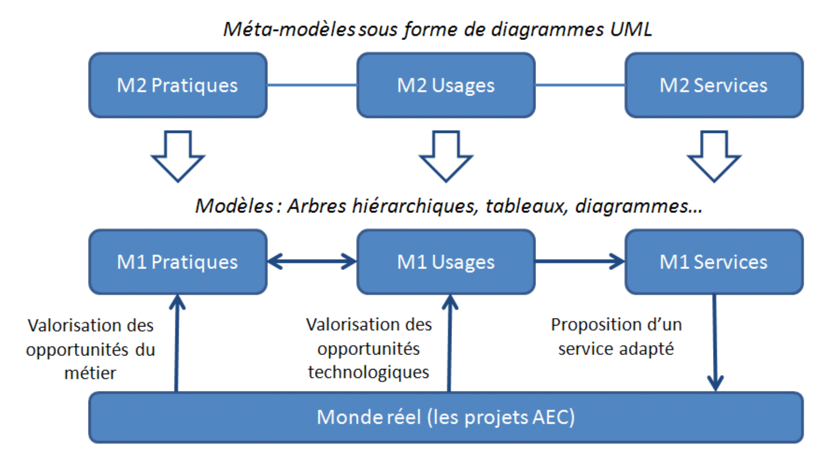



*Figure 83. Une méthode dirigée par les modèles et favorisant l'innovation*

[29] http://www.eclipse.org/atl/

Nous parlons de valorisation des opportunités métier lorsque la méthode permet d'utiliser les analyses d'un domaine comme source d'innovation dans le développement de solutions informatiques adaptées. Le RUP est un processus de référence de ce type.

Parallèlement, des expériences antérieures ont montré que la démarche de conception peut également être initiée par une opportunité au niveau technologique. C'est souvent le cas dans le domaine des IHM avec le développement de nouveaux modes d'interaction qui ouvrent le champ des possibles en termes d'usage (par exemple l'interface gestuelle « Kinect »). Cependant, ces moyens ne sont que rarement utilisés pour répondre à de réels besoins. Mal exploités, ils ne génèrent alors rien au-delà de l'engouement initial que l'on confère à l'innovation technologique. Avec la méthode proposée dans cette thèse, nous voulons favoriser ce « push technologique », en identifiant les pratiques susceptibles d'être outillées et améliorées par les technologies. Dans un deuxième temps nous pourrons voir comment les dites technologies, guidées par ces pratiques, pourront offrir des services adaptés.

Cette double approche caractérise les démarches SOA (Architecture Orientée Services). Comme le dit (Idoughi & Kolski 2009) citant (Avignon et al. 2002), « une démarche SOA peut être motivée et pilotée par les métiers, qui imposent à l'informatique ou au système d'information d'une organisation d'évoluer suite aux changements incessants et fréquents des besoins métiers. Elle peut aussi être pilotée par les technologies, où il s'agit d'offrir des opportunités aux différents métiers à travers l'émergence de nouveaux besoins suite à l'évolution de l'offre technologique (Internet, téléphonie mobile…) ».

Franchir l'écart entre les domaines

Notre positionnement par rapport aux approches SOA favorisant à la fois le « push métier » et le « push technologique » nous permet de créer un lien entre le métier et la technologie. Les concepts de pratique et d'usage supportent ces deux dimensions. Ils s'inscrivent ainsi dans une méthode de conception de services collaboratifs qui devrait permettre de pallier à « l'échec de la solution collecticiel global », non pas seulement par des grands principes, mais bien par une démarche scientifique de conception.

Cette méthode que nous nommons « PUSH » (pour « Practice and Usage based Service enHancement », traduit en français par « amélioration des services sur la base des pratiques et usages »), supporte la collaboration entre les experts du domaine AIC et les experts du génie logiciel (Santos et al. 2012). Dans la section suivante, nous nous interrogeons sur l'intervention de ces différents acteurs d'un projet de développement de services en fonction de leur activité. Nous présentons le support documentaire de cette collaboration : le cahier d'exigences.

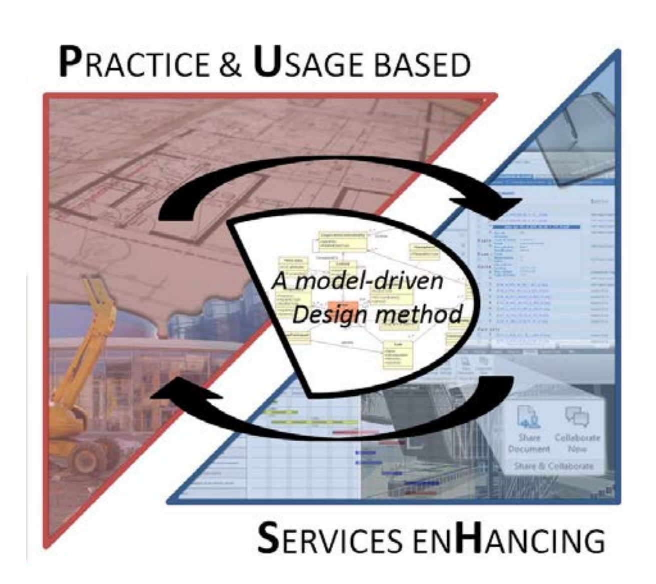



*Figure 84. Représentation abstraite de la méthode PUSH*

### *1.25.2 Le cahier « cahier d'exigences »*

D'après (Paetsch et al. 2003), « un bon document d'exigences est non-ambigu, complet, correct, consistant, concis et réalisable. En fonction de la relation entre le client et le fournisseur, la spécification des exigences peut être contractuelle ».

Le « cahier d'exigences pour la conception et spécification de services ICT adaptés » a été créé pour y consigner les modèles et donc les différents points de vue exprimés au cours de l'application de la méthode PUSH. Il est composé de deux documents distincts : le mode d'emploi et le formulaire de définition des exigences[30]. Le mode d'emploi donne les éléments nécessaires à la compréhension et l'édition du formulaire (voir annexes). Il résume notamment la description des concepts détaillée dans les chapitres suivants. Le formulaire est un document dont l'édition est structurée et guidée. Nous en avons réalisé le modèle avec les fonctionnalités de développement internes à Microsoft Word comme :

- la création champs d'insertion d'image afin d'importer les modèles graphiques,
- la création de champs de choix multiples et de saisie organisés dans des tableaux pour éditer les modèles textuels,
- des champs de saisie supplémentaires qui permettent d'ajouter des commentaires et informations additionnelles.

La première page comprend le titre donné au service à développer, les acteurs impliqués dans la conception de ce service (selon leur rôle parmi l'utilisateur, l'analyste métier, concepteur et le développeur), et un suivi des versions (avec numérotation, date, et description des changements).

Le plan du formulaire suit ensuite la trame de notre approche, comprenant ainsi une partie 1 : « pratiques métier », une partie 2 : « usages » et une partie 3 : « spécification du service ». Chacun des

[30] Le formulaire et les modèles qu'il comprend sont en anglais afin d'être plus facilement publiés. Le mode d'emploi est cependant en français.

chapitres suivants détaillant ces trois concepts et leur modélisation, se terminera par une section illustrant l'insertion des modèles dans le cahier d'exigences.

Enjeux et utilisation du document

Dans la composition de notre document, nous avons souhaité pallier à deux problèmes (également reconnus dans (Paetsch et al. 2003)) :

- la constitution d'un tel document dans les méthodes agiles est souvent jugée trop compliquée, voire infaisable,
- les modèles produits sont éphémères et ne « rentrent pas dans la documentation du système ».

Le cahier d'exigences que nous proposons se veut simple à lire et à éditer de par la présentation d'information sous forme graphique et structurée dans un format de document court (environ 10 pages). Ce document a été pensé comme une source d'informations qui alimente les travaux de développement lors de la proposition de services adaptés au contexte métier. Il doit permettre également la justification de ces développements à posteriori. Les enjeux de ce document sont le transfert (vers les développeurs) et la traçabilité des choix.

Il est destiné :

- Aux analystes métier pour consigner les modèles des pratiques étudiées auprès des futurs utilisateurs. L'équipe (actuelle ou une future) de développement s'y référera en cas de doute sur les « origines métier » des développements.
- Aux concepteurs en collaboration avec les analystes pour consigner les modèles des usages imaginés et à outiller. Les développeurs étudieront les choix formalisés pour proposer une solution qui corresponde. Ici encore l'équipe (actuelle ou une future) de développement pourra s'y référer en cas de doute sur les choix de développements.
- Aux développeurs avec les concepteurs pour spécifier et consigner les modèles de l'architecture du service proposé. Cette formalisation s'avèrera très utile pour la pérennisation du service en cas de changement d'équipe de développement.

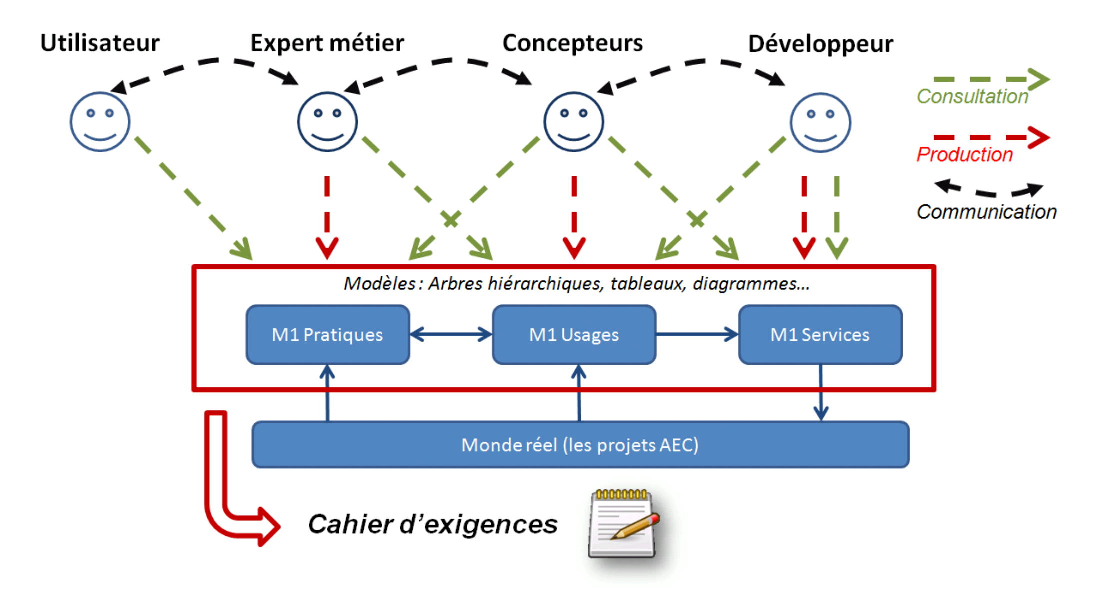



*Figure 85. Génération du cahier d'exigences à partir des modèles créés*

## 1.26 Conclusion

Nous proposons une méthode pour concevoir de services informatiques adaptés aux pratiques métiers d'un projet de conception/construction architecturale. Elle est basée sur la description de trois concepts :

- les pratiques métiers identifiées : c'est ce qu'on appelle le point de vue organisationnel dont est responsable l'expert métier,
- les usages associés choisis : c'est le point de vue opérationnel, exprimé par le concepteur,
- les services à spécifier : c'est le point de vue fonctionnel attribué aux développeurs.

Il s'agit de tirer profit du contexte académique et professionnel de cette étude pour définir ces concepts, puis de fournir les outils pour les décrire. Ces descriptions sont conformes à un méta-modèle : le Méta-Modèle des Services Adaptés (MMSA). Elles sont consignées dans un cahier d'exigences qui assure le suivi du processus et la traçabilité des choix pendant et après le projet de développement.

Les chapitres suivants (Chapitres 9, 10 et 11) détaillent chacun des trois points de vue, les concepts associés, leur méta-modèle et leur description selon des formalismes adaptés.

# Chapitre 9 – La modélisation des pratiques : le point de vue organisationnel

Ce chapitre est composé de trois sections. La première présente le concept de pratique, tel que nous le définissons pour représenter les comportements de professionnels dans leur contexte de travail collaboratif et observés lors d'une analyse métier. La caractérisation de ces pratiques est liée au domaine AIC qui est le domaine d'application de cette étude. La deuxième section présente la construction du « méta-modèle des pratiques métier » (MMPM) qui supporte cette caractérisation par différents concepts. Le formalisme utilisé pour le MMPM est le diagramme de classes UML. La troisième section montre ensuite comment ce méta-modèle permet la modélisation des pratiques métiers, notamment grâce à un éditeur dédié. Elle conclut ce chapitre par l'intégration de ce modèle dans un cahier d'exigences. Le framework pour la génération de l'éditeur à partir du MMPM et la nature du cahier d'exigences ont tous deux été introduits dans le chapitre précédent.

## 1.27 Définitions et concepts

### *1.27.1 Les pratiques : un nouveau découpage de l'activité de projet*

Au cours du 0 (voir 1.1.3) nous avons décomposé l'activité de projet dans le domaine AIC en phases et sous-phases (Figure 86).

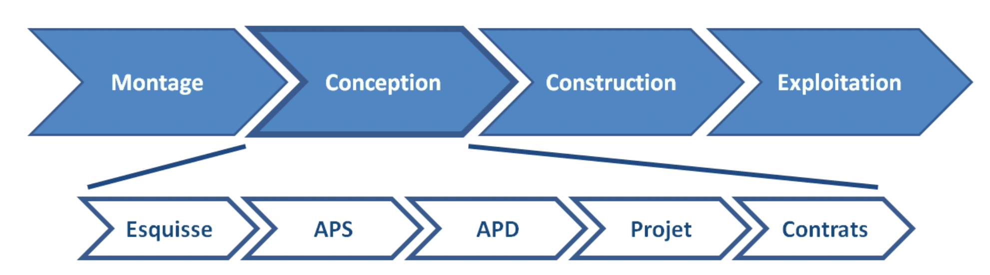



*Figure 86. Découpage en phases et sous-phases d'un projet AIC*

Cependant le passage d'une phase à une autre reste flou, il est d'ailleurs souvent admis qu'elles se chevauchent, le début d'une phase ne marquant pas tout à fait la fin de la précédente. De plus, même si l'on connait les objectifs généraux relatifs à chaque phase, il est difficile de se représenter le déroulement exact de celles-ci, la combinaison des tâches étant particulière à chaque opération. Comme on peut le lire dans (Bignon et al. 2009), « la lisibilité des processus en cours n'est pas aisée et elle est peu partagée ; la traçabilité des évolutions du projet et des décisions est difficile ; les flux d'informations sont souvent interrompus… ».

Pour pallier à ces limites dans la compréhension des processus métiers, nous nous orientons vers un nouveau découpage des processus de projet AIC, d'abord assez générique, puis en opérant plusieurs raffinements, afin d'appréhender les spécificités et variabilités du processus de chaque projet. Nous cherchons alors à :

- exprimer des objectifs précis, relatifs à des acteurs particuliers et pouvant varier d'un projet à un autre,
- exprimer des flux d'information qui concernent plusieurs phases voire qui peuvent être interrompus et repris au cours du projet.

Dans notre étude, et pour cette dimension « métier », nous adoptons le terme de « pratique ». Nous essayons à travers ce concept de pratique de prendre de la distance par rapport aux descriptions de processus trop « figés », cherchant à décrire le caractère par nature flexible de l'activité de projet AIC. Nous retrouvons d'ailleurs ce concept dans plusieurs études que nous évoquons ci-dessous.

Pratique : définition

Nous avons introduit ce concept en quelques lignes dans le chapitre 3 de ce mémoire. La pratique peut être littéralement définie comme l'exercice d'un métier, une manière de travailler, un comportement habituel avec une finalité. En d'autres termes, c'est un comportement adopté par une plusieurs personnes dans le but d'atteindre un objectif métier. D'après (Schmidt & Wagner 2004), le travail coopératif dans un projet AIC ne peut pas être défini comme un processus complet car les acteurs agissent de manière « hautement distribuée à travers de nombreuses pratiques et artefacts ».

Pour (Marjanovic et al. 2007) les outils de TCAO sont souvent « conçus pour supporter des tâches individuelles sans les placer dans le contexte d'un processus figé ». Les auteurs ajoutent que « le champ du TCAO considère des activités hors des limites organisationnelles formelles, gouvernées par des règles et politiques ». Ils introduisent les « processus métiers orientés pratiques », qu'ils distinguent des « processus orientés procédures ». Les pratiques sont, dans le domaine de « la gestion des connaissances » (knowledge management), relatives à une connaissance tacite (alors que les procédures relèvent d'une connaissance explicite). Les processus orientés pratiques servent à modéliser « la coordination au travers de tâches à bas niveau de granularité », dépendantes des décisions humaines et peu prédictibles. Ces « tâches » se distinguent donc des tâches des processus métiers classiques (orientés procédure) car elles émergent avec l'exécution du processus. Nous retiendrons également de cette étude le rôle particulier de « l'expert métier » qui analyse ces processus et est à même de les comprendre et les formaliser.

D'autres concepts similaires sont traités dans la littérature. Par exemple, les « mécanismes de coordination » (Schmidt & Simone 1996) sont des « protocoles qui déterminent et médiatisent la coordination d'activités distribuées afin de réduire la complexité du travail » (Tellioglu 2006). Dans (Sandkuhl 2010), les « patrons d'information » (Information Supply Pattern) ont pour but de « capturer la connaissance sur des solutions prouvées pour en faciliter la réutilisation ». De tels patrons concernent « des problèmes d'informations récurrents qui surviennent pour des rôles et dans des situations spécifiques et présentent des solutions conceptuelles à ces problèmes ». Ils sont définis par un nom, un contexte organisationnel, un problème, une solution conceptuelle, les effets de cette solution et enfin les actions liées à l'information produite.

(Bourguin & Derycke 2005) relèvent la nécessité de « transformer, adapter les pratiques et les méthodes de travail, pour satisfaire les besoins évolutifs de l'organisation ». Nous retrouvons également ici le caractère flexible souligné précédemment. Ils font émerger les apports de la Théorie de l'Activité (TA) pour la caractérisation des pratiques, caractérisation que nous adopterons.

Rappelons que la TA décrit une structure hiérarchique composée de 3 niveaux : l'activité, l'action et l'opération. Considérant le projet et ses phases comme une activité structurée, nous pouvons définir les pratiques comme des actions, « individuelles ou collectives, mais toujours dirigées vers un but conscient » (Bourguin & Derycke 2005). Une action est en effet définie comme fortement dépendante de l'activité de laquelle elle découle (N.B. elle peut découler de plusieurs activités). Elle est planifiée en fonction d'un objectif et d'une finalité. Enfin, elle génère de la connaissance.

Tableau 11. Notre approche par rapport aux concepts de la Théorie de l'Activité

| Théorie de l'activité | Concepts manipulés | Notre approche |
|---|---|---|
| L' activité | Besoins généraux | Le projet |
| Les actions | Buts conscients | Les pratiques |

Pour comprendre la nature de ces pratiques dans un projet AIC nous en faisons une première description générale répartie en 11 familles de pratiques.

### *1.27.2 Les familles de pratiques (ou pratiques collectives génériques)*

Le domaine des projets de conception/construction AIC a pris ces dernières années un tournant important, faisant l'objet de nombreuses évolutions. Parmi celles-ci, il faut tout d'abord prendre en compte le contexte économique, sans cesse plus contraignant. La performance environnementale est également devenue une dimension essentielle et complexe, poussant les concepteurs mais aussi les techniciens à innover, à se renouveler dans leurs pratiques. Enfin, il faut s'assurer que la qualité architecturale du bâtiment n'en pâtisse pas.

Afin d'augmenter la performance environnementale des bâtiments construits (et rénovés) mais aussi afin d'évaluer ces performances, des institutions proposent des guides et référentiels

destinés aux professionnels du secteur. Leur analyse nous a permis d'identifier et définir 5 objectifs principaux à atteindre. Les référentiels analysés sont[31] : HQE (France), BREEAM (Royaume-Uni), MINERGIE (Suisse) et DGNB (Allemagne). Ce dernier, le référentiel allemand DGNB, est le plus précis et le plus complet. De la revue des ces référentiels environnementaux, nous avons identifié les cinq objectifs suivants (Zignale, Sylvain Kubicki & Gilles Halin 2011) :

- Assurer la qualité du bâtiment : cela inclut les choix conceptuels et les choix techniques. La qualité architecturale (comme l'agencement des espaces, les relations intérieur/extérieur, etc.) est également importante. Cet objectif concerne la fiabilité structurelle ou encore la qualité des ouvrages dans le temps. Il comprend aussi des exigences plus spécifiques comme un impact limité sur l'environnement par la réduction des consommations énergétiques, par l'utilisation de matériaux écologiques…
- Assurer le confort des habitants : les choix conceptuels et techniques sont également questionnés au regard du confort thermique, acoustique et visuel. Le confort est aussi relatif aux aspects d'accessibilité et de sécurité.
- Assurer le rendement économique : les acteurs et particulièrement les concepteurs doivent contrôler le budget du projet en maitrisant les coûts qu'impliquent leurs choix. Ils chercheront aussi à rendre le bâtiment construit plus rentable dans le temps, en assurant sa flexibilité et son adaptabilité à plusieurs usages (surtout pour les bâtiments publics).
- Assurer la qualité du site : en accordant de l'importance à sa localisation et aux divers risques liés, à la qualité de l'environnement, à l'accès aux transports, aux services et infrastructures présentes…
- Assurer la qualité socioculturelle du projet en mesurant et en prenant en compte son impact sur la population environnante, sur les administrations… (particulièrement pour les projets de grande envergure ou à caractère exceptionnel).

Nous avons cherché à identifier les pratiques généralement mises en place pour atteindre ces cinq objectifs au cours d'un projet. Ce travail s'est déroulé sur la base de deux brainstormings avec respectivement 3 et 2 architectes. Pour chaque objectif il était question de recueillir leurs expériences, habitudes et anecdotes. Nous avons synthétisé ces informations en 11 familles de pratiques distinctes.

La description qui suit comprend, pour chaque famille de pratique, un texte explicatif décrivant son déroulement puis un tableau comprenant un résumé des rôles impliqués, des documents utilisés et des activités (phases et tâches) dans lesquelles la pratique s'inscrit. Nous nous sommes inspiré de l'ouvrage « 140 séquences pour mener une opération de construction : des études préalables à l'achèvement de l'ouvrage, actions techniques et démarches administratives » (Armand 1997) pour mener cette synthèse.

PC1 : choix et évaluation du site

**Description :** le Maître d'ouvrage (MO) définit la nature de l'ouvrage et justifie le choix du terrain de par l'impact (physique et social) qu'aura l'ouvrage, mais aussi en analysant le contexte urbain (équipements, services…). Il s'interroge également sur les équipements complémentaires nécessaires. Le site doit faire l'objet de plusieurs analyses de la part du Maître d'Ouvrage (nuisances, risques d'inondation, possibilités d'alimentation et d'évacuation, nature du sol, accès…) qui se déroulent avant et pendant les premières phases de la conception. Il est également nécessaire de recueillir auprès des services adéquats (mairie ou administration communale,

[31] http://assohqe.org/hqe/; http://www.breeam.org/; http://www.minergie.com/home_en.html; http://www.dgnb.de/_en/

services techniques, opérateur de téléphonie, fournisseurs d'électricité, de gaz, compagnie des eaux…) l'ensemble des renseignements concernant la viabilité du terrain, et notamment les accès possibles et les différents réseaux d'alimentation et d'évacuation.

Tableau 12. Éléments de caractérisation de la PC1

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'ouvrage<br>Maître d'œuvre<br>Ingénieurs (géologues…)<br>Administration (commune) | Plans APS<br>Plan Local d'Urbanisme<br>Rapports d'études | Tâches d'expertise et coordination<br>Phase Montage (études préalables)<br>Phase Conception (APS) |

PC2 : choix de la maîtrise d'œuvre

**Description :** le maître d'ouvrage engage la Maîtrise d'œuvre qui devra concevoir l'ouvrage. Dans le cas de projets de grande envergure, la consultation passe par les phases d'un appel d'offres : appel public, dossier de consultation (clauses de l'engagement, du CCAP, CCTP), critères de choix sur les compétences, choix et notifications des marchés. Chaque titulaire de marché renseignera les buts et performances à atteindre, les techniques de base à utiliser, les moyens en personnel et matériel à mettre en œuvre, le niveau du prix des prestations, un découpage en phase. Le maître d'ouvrage peut ainsi évaluer leur capacité à mener à bien son projet dans des conditions et avec des performances optimales.

Tableau 13. Éléments de caractérisation de la PC2

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'ouvrage<br>Maître d'œuvre | Dossier de consultation<br>Plans APS<br>Documents administratifs et contractuels | Tâches conception et coordination<br>Phase d'études préalables<br>Phase Conception APS (si appel d'offres) |

PC3 : détermination des objectifs

**Description :** il appartient au Maître de l'ouvrage de définir le programme de l'opération. Il définit le processus de construction qui comprend des éléments quantifiables et techniques ainsi que des éléments fonctionnels, qualitatifs et évolutifs. L'assistance à la Maîtrise d'ouvrage a pour rôle de l'assister dans cette tâche. S'il veut prétendre à l'attribution d'un label environnemental, il peut se référer à des experts du domaine. Il fixe ainsi les objectifs avec lesquels devra travailler la Maîtrise d'œuvre lors de la conception de l'ouvrage. Lors de petits projets, le Maître d'ouvrage peut n'avoir que très peu d'informations à donner, c'est le Maître d'œuvre qui fixe alors ces objectifs.

Tableau 14. Éléments de caractérisation de la PC3

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'ouvrage<br>Assistance à la MO<br>Maître d'œuvre<br>Experts | Programme<br>Informations et directives techniques | Tâches conception et coordination<br>Phase d'études préalables<br>Phase Conception APS (si appel d'offres) |

PC4 : détermination et gestion du budget

**Description :** le Maître d'ouvrage arrête l'enveloppe financière lors du programme. Les documents financiers du projet, depuis la proposition de prix jusqu'au solde de tout compte, constituent l'une des composantes majeures du marché entre le MO et les organismes engagés. Les documents élaborés au cours du projet (plans, descriptifs, avant-métré) qui serviront à la consultation des entreprises permettent d'établir un coût prévisionnel des travaux par corps d'état (ou par lot). Il y a une demande très forte auprès des économistes de la construction qui sont capables de faire le pont entre les bureaux d'études et les entreprises. Sa tâche est de définir l'enveloppe financière d'un projet en maîtrisant les aspects matériaux et énergie dans une construction tout en respectant les règles imposées.

Tableau 15. Éléments de caractérisation de la PC4

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'ouvrage et assistants<br>Maître d'oeuvre<br>Comptable, économiste<br>Tout contractant | Enveloppe et documents financiers<br>Plans et descriptifs liés à la conception | Tâches de coordination<br>Toutes phases |

PC5 : conception et compte-rendu de la conception

**Description :** l'esquisse constitue la première étape de la réponse architecturale et technique au programme. C'est la formalisation graphique des premiers choix du concepteur. Elle instaure un dialogue entre Maître d'ouvrage et Maître d'œuvre. Le concepteur extrait du programme les éléments nécessaires et les formalise en aide-mémoires, organigrammes de services, ou fonctions... Les principaux éléments à retenir sont les objectifs du Maître d'ouvrage, les considérations touchant à l'intégration de l'ouvrage dans son environnement et à l'organisation spatiale. L'organisation générale des volumes doit être affinée en fonction du programme du Maître d'ouvrage. Le choix de la structure doit se faire par la collaboration du Maître d'œuvre et de l'ingénieur structure, avec l'accord final du contrôleur technique. Cette phase comprend aussi les choix relatifs au confort thermique et acoustique. Une notice descriptive résume les principaux choix effectués, ainsi que les principales options retenues. C'est sur la base de ces descriptifs, en complément des plans de l'avant-projet détaillé (APD), et après accord du Maître d'ouvrage, que les cahiers des clauses techniques particulières (CCTP) seront rédigés. La demande de permis de construire se fait auprès de la mairie. Le MOE doit assistance au MO durant toute la durée de l'instruction du permis de construire.

Tableau 16. Éléments de caractérisation de la PC5

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Ingénieurs<br>Contrôleur technique<br>Maître d'ouvrage et assistants | Programme<br>Plans de conception, modèles 3D…<br>Documents techniques<br>Documents administratifs | Tâches de conception et de coordination<br>Phase conception (de l'esquisse à l'envoi des documents pour exécution) |

PC6 : Choix des entreprises de construction

**Description :** Le dossier de consultation est remis aux entreprises par le Maître d'œuvre afin qu'elles puissent établir leurs offres en vue de l'attribution des marchés. Il contient les cahiers des clauses administratives et techniques particulières (le CCAP et le CCTP), les dossiers des plans d'architecture et techniques, les pièces financières et éventuellement le rapport de sol, les plans parcellaires, les plans des réseaux, les notes de calculs etc. Les entreprises soumettent alors leur dossier lors de l'appel de candidature de l'appel d'offres. Les documents nécessaires à la consultation étant établis, il reste à organiser la remise par le Maître d'œuvre de ces documents et à en assurer la diffusion auprès des entrepreneurs. Après avoir établi le dossier des pièces qui lui sont nécessaires pour faire acte de candidature, l'entrepreneur doit transmettre son offre. Qu'il soit engagé directement ou à la suite d'un appel d'offres, l'entrepreneur retenu doit être reconnu comme compétent pour réaliser l'ouvrage ou un lot de l'opération. Il devra satisfaire à certains critères vérifiables tels que le prix des services fournis ou les qualifications (obtenues par des formations reconnues). Il pourra également fournir des garanties supplémentaires quant à la qualité et la fiabilité de l'entreprise, au travers par exemple de la qualité des prestations antérieures ou des recommandations d'autres professionnels.

Tableau 17. Éléments de caractérisation de la PC6

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Entreprises<br>Maître d'ouvrage et assistants | Plans divers<br>Documents administratifs<br>Pièces financières | Tâche de coordination<br>Phase conception (à partir de la réalisation de plans communicables) |

PC7 : évaluation de la conception et compte-rendu

**Description :** lors des dernières phases de la conception, il est nécessaire d'en évaluer certains critères comme ceux relatifs à la santé, la sécurité, l'hygiène, l'accessibilité… et des spécialistes de chaque domaine sont impliqués pour mener leur évaluation. Dans des cas particuliers dont les projets d'architecture environnementale ou durable, les experts évaluent également les performances énergétiques, acoustiques ou lumineuses de l'ouvrage. En fonction de la nature du projet et des objectifs fixés au début de son cycle de vie, les résultats de ces évaluations conditionnent les choix définitifs ainsi que l'exécution du projet.

Tableau 18. Éléments de caractérisation de la PC7

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Experts<br>Maître d'ouvrage et assistants | Plans d'exécution<br>Modèles 3D<br>Documents techniques<br>Rapports d'expertises | Tâches d'évaluation et de coordination<br>Phase conception (avant exécution) |

PC8 : organisation de réunions et compte-rendu

**Description :** pendant le montage et la conception, un certain nombre de rencontres sont organisées entre le maître d'ouvrage, le maître d'œuvre mais aussi les experts, le personnel administratif, les élus… La personne qui tient le rôle de coordinateur organise ces réunions et formalise un rapport qui contiendra tous les éléments importants évoqués (ordre du jour, objectifs, avancement depuis la dernière réunion, décisions…). Lors du chantier et de sa préparation, le coordinateur doit organiser, en concertation avec les entreprises, un ensemble de réunions de coordination technique. Le responsable est celui qui est défini comme animateur de la réunion. On devra toujours y trouver une des personnes suivantes : le maître d'œuvre, le coordinateur désigné pour cette mission, l'entrepreneur titulaire d'un marché unique, le mandataire ou entrepreneur général. Avant une réunion de chantier, il est souhaitable que le MOE (éventuellement en compagnie du coordinateur) effectue une visite de chantier, même rapide, afin de visualiser l'avancement du chantier, de noter les travaux défectueux ou non conformes, de déceler les difficultés qui risquent de resurgir, afin de conduire la réunion avec un maximum d'efficacité. En cas de contentieux le compte-rendu de chantier constituera toujours une pièce importante d'expertise.

Tableau 19. Éléments de caractérisation de la PC8

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Experts<br>Maître d'ouvrage et assistants<br>Entreprises | Tout document servant de base à la discussion<br>CR de réunion | Tâches d'évaluation et de coordination<br>Toutes phases (plus formelles et structurée pendant le chantier) |

PC9 : préparation et gestion du chantier

**Description :** l'organisation du chantier doit prendre en compte la mise en place d'un planning dans lequel seront réparties toutes les interventions de construction. Ce planning pourra et devra être ajusté en cas de changements, de retards… Cela implique également la livraison et le stockage des matériaux, la mise en place et la gestion des engins, la gestion de la sécurité. Les entreprises qui interviennent sont chargées d'exécuter leurs tâches dans les délais impartis en respectant l'ordonnancement décrit dans le planning. C'est le coordinateur qui s'assure du bon déroulement du chantier et prend note des problèmes rencontrés. Le maître d'œuvre se doit de prendre en compte les remarques des exécutants en cas de modifications à apporter aux plans d'exécution. Beaucoup d'ajustements sur chantier sont informels : ces modifications de petite

envergure ne sont pas consignées dans les plans ou les comptes-rendus, mais sont effectuées à partir de remarques orales. Le Maître d'ouvrage peut également intervenir en cas de litiges.

Tableau 20. Éléments de caractérisation de la PC9

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Coordinateur<br>Maître d'ouvrage et assistants<br>Entreprises | Plans d'exécution<br>Documents techniques<br>Planning<br>CR de chantier | Tâches de construction, dévaluation et de coordination<br>Phase chantier |

PC10 : évaluation de l'exécution et compte-rendu

**Description :** l'évaluation du bâtiment en cours d'exécution et juste avant la livraison revient au coordinateur et aux divers experts. Le coordinateur relève les malfaçons en cours de chantier et les transmet aux entreprises responsables (ils pourront être sujet de discussion lors des réunions de chantier). Les experts en sécurité et santé vérifient que les normes sont respectées pendant la mise en œuvre (port du casque, filets de sécurité, etc…) et lors de la livraison (garde-corps, sorties de secours…). Les experts en thermique ou acoustique effectuent les tests nécessaires pour évaluer les performances du bâtiment, en particulier en cas d'attribution d'un label. Le Maître d'ouvrage doit être informé du déroulement et des résultats de chaque intervention.

Tableau 21. Éléments de caractérisation de la PC10

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Coordinateur<br>Maître d'ouvrage<br>Entreprises<br>Experts | Plans d'exécution<br>Documents techniques<br>Référentiels, normes<br>CR d'expertise<br>CR de chantier | Tâches de construction, d'évaluation et de coordination<br>Phase construction (pendant le chantier et avant la livraison) |

PC11 : implication des usagers

**Description :** L'implication des usagers n'est pas systématique, surtout lorsqu'ils ne sont pas directement les maîtres de l'ouvrage. Le cas le plus typique est l'ajustement de certaines décisions de conception ou d'aménagement (pièces, mobilier, équipements…), notamment dans des opérations de construction à grande échelle (lotissement, logement collectif). En ce qui concerne les bâtiments publics, l'implication des futurs usagers durant le projet consiste essentiellement à la communication d'informations. Une fois le bâtiment conçu et au regard des préoccupations environnementales, son usage doit être encadré par un ensemble de « bonnes pratiques » qui doivent être transmises aux usagers (réduction de la consommation d'énergie, gestion des déchets, logique dans l'utilisation des équipements de chauffage/ventilation…).

Tableau 22. Éléments de caractérisation de la PC11

| Acteurs impliqués | Documents utilisés | Activités |
|---|---|---|
| Maître d'œuvre<br>Maître d'ouvrage<br>Usagers | Posters<br>Articles<br>Règles et normes<br>Documents techniques | Tâches de coordination<br>Toutes phases |

### *1.27.3 Enjeux dans l'instrumentation des pratiques métiers et approches de modélisation*

Comme introduit précédemment, l'objectif de cette étape de notre méthode est de concevoir des services supports aux activités collectives de projet dans le domaine AIC. Ce premier travail d'analyse métier sur les familles de pratiques nous permet d'appréhender de manière globale l'activité de projet et de la diviser en sous-ensembles compréhensibles. Cependant, pour mener une analyse métier qui servira de base à la conception de services collaboratifs adaptés, il est nécessaire d'aller au-delà de ces grands principes : il nous faut pouvoir décrire finement les pratiques observées en situation réelle de projet.

Dans notre approche, nous distinguons deux types de pratiques :

- les **pratiques collectives**, relatives aux objectifs d'un groupe d'acteurs (aux minimum deux acteurs collaborant),
- les **pratiques individuelles**, effectuées individuellement par chacun des acteurs de ce groupe pour atteindre l'objectif identifié en commun.

Sur la base de l'analyse des onze familles précédentes, notre démarche scientifique se poursuit en proposant une conceptualisation de ces pratiques à travers un méta-modèle dont le principe d'instanciation assurera une description structurée et uniforme, d'un projet de conception à un autre (tel qu'introduit dans la définition de notre méthodologie, cf. 1.24). Les travaux précédents sur le Méta-Modèle du Contexte Coopératif (MMCC) décrit en 1.1, définissent déjà un ensemble de concepts servant de base à cette description.

Nous utilisons les diagrammes de classe UML pour modéliser les concepts utiles à la description de ces pratiques, leurs attributs et les relations sémantiques qui les relient. Le Méta-Modèle des Pratiques Métier (MMPM) est ainsi construit à partir de deux méta-modèles :

- le Méta-Modèle des Pratiques Collectives (MMPC),
- le Méta-Modèle des Pratiques Individuelles (MMPI).

## 1.28 Le Méta-Modèle des Pratiques Métier

### *1.28.1 Les pratiques collectives*

Nombreuses sont les pratiques collectives qui peuvent être identifiées lors de l'observation d'un projet. Par exemple, la pratique suivante est extraite de la **famille** 5: « Le choix de la structure doit se faire par la collaboration de l'architecte et de l'ingénieur structure, avec l'accord final du contrôleur technique ». Un **nom** pour identifier la pratique peut être isolé de cette **description**, comme par exemple « choix de structure ».

Nous pourrions décrire de cette manière toutes les pratiques identifiées. Ainsi, le nom, la description et la famille seront les attributs permettant de caractériser une pratique collective. La classe « Pratique collective » dans le diagramme de classes UML suivant (Figure 87) représente cette caractérisation. La détermination du nom et de la description se fait par saisie de texte alors que la famille est choisie dans une énumération (voir le paragraphe « la modélisation et ses outils » en 1.24.1).

Il s'agit ensuite de définir les acteurs, les artefacts et les activités impliqués dans la pratique. Certaines des possibilités auront peut-être déjà été identifiées dans la description comme par exemple ici les acteurs : l'architecte, l'ingénieur structure et le contrôleur technique. Ces concepts ont été définis au cours des études précédentes du laboratoire MAP-CRAI dans le MMCC (Méta-Modèle du Contexte Coopératif). Le MMPC que nous proposons a pour but de modéliser l'implication de ces concepts dans une pratique collective.

Il faut souligner que les énumérations décrites dans les méta-modèles suivants ne sont pas toujours exhaustives et pourraient être complétées. À cet effet, un champ « autre » est inclus comme choix lors de l'instanciation. Il faut alors renseigner l'information voulue sous forme de remarque et compléter si besoin le méta-modèle.

La classe acteur est spécifiée par deux classes : l'acteur simple et le groupe d'acteur.

- Le type d'acteur simple (« type_acteur(simple) ») est relatif à son métier. Les choix possibles sont : architecte, urbaniste, ingénieur structure, ingénieur sécurité, ingénieur santé, chef d'entreprise, maçon, électricien, comptable, secrétaire, autre…
- Le type de groupe (« type_acteur(groupe) ») est relatif au statut administratif du groupe (agence/bureau d'études, entreprise, laboratoire, organisation publique, autre…).
- Tout acteur (simple ou groupe) joue un rôle dans le projet, c'est ce qui a été défini comme le rôle organisationnel (« type_rôle »). On distingue les rôles suivant : Concepteur, Dessinateur/Graphiste, Coordinateur, Maître d'ouvrage, Constructeur, Économiste, Conseiller, Expert, Administration, autre....
- L'attribut dénomination servira à apporter des distinctions entre acteurs comme par exemple entre « le dessinateur des façades » et « le dessinateur des plans ».

Sous le concept d'activité, on retrouve :

- les tâches de type (« type_activité(tâche) ») conception, construction et coordination,
- les phases de type (« type_activité(phase) ») préparation, conception, exécution et livraison,
- ainsi que la caractérisation du projet lui-même par son type (« type_activité(projet) ») :logement individuel, logement collectif ou bâtiment public ainsi qu'une éventuelle certification visée (HQE, BREEAM, MINERGIE, DGNB, autres, pas de certification).

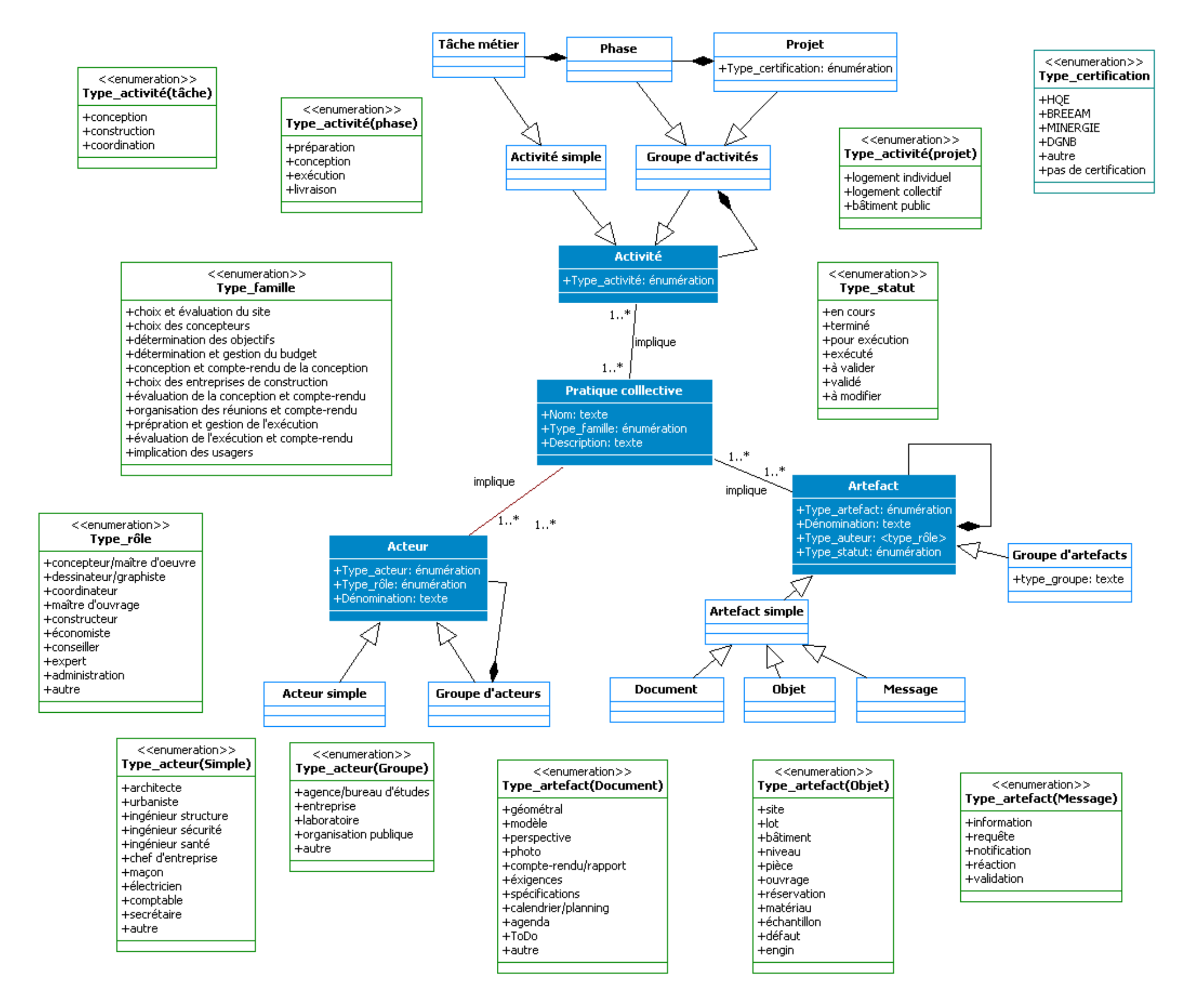



*Figure 87. Méta-Modèle de la pratique collective (MMPC)(concepts principaux : fond bleu, spécifications :contours bleus, énumérations : contours verts)*

Enfin, nous distinguons trois artefacts pouvant être typés (« type_artefact(document-objet-message) ») et caractérisés par leur auteur et leur statut :

- Les documents sont les produits physiques ou numériques des tâches de conception et de coordination (voir 1.1.2). Les « type_artefact(document) » identifiés sont : géométral, modèle, perspective, photo, compte-rendu/rapport, exigences, spécifications, calendrier/planning, agenda, « ToDo ».
- Les objets sont relatifs aux ouvrages de construction (voir également 1.1.2) et sont définis à plusieurs échelles. Ainsi, les « type_artefact(objet) » identifiés sont (du plus grand au plus petit) : le site, le lot, le bâtiment, le niveau, la pièce, l'ouvrage, la réservation, le matériau, et l'échantillon. On y ajoute le défaut (notion de malfaçon sur un ouvrage) et le véhicule de chantier.
- Les messages, qu'ils soient formalisés ou pas, sont les traces de toute communication qui s'instaure au cours des différentes tâches. Parmi les « type_artefact(message) », on distingue les messages d'information classiques d'autres types plus spécifiques comme les requêtes (qui impliquent un retour, une réponse), les notifications (pour avertir), les réactions (qui ciblent quelque chose en particulier) et enfin les validations (qui permettent de donner son approbation sur quelque chose).

Nous n'avons pas typé les groupes d'artefact, la nature pouvant être très variée selon le contenu (ensemble des plans, ensemble des rapports, permis de construire, cahier des charges…). L'auteur d'un artefact (simple ou groupe) est un acteur du contexte coopératif tel que nous l'avons défini précédemment. Nous l'identifions alors par son rôle, c'est pourquoi dans le méta-modèle l'attribut « type_auteur » est relié à l'énumération « type-rôle ».

Le statut d'un artefact est un attribut important dans la modélisation des pratiques car il permet de suivre l'état d'avancement des tâches qui leur sont liées et donc du projet. Le statut le plus répandu dans un projet de construction est le statut « bon pour exécution » qui est attribué aux plans à envoyer sur le chantier pour exécution des travaux. Il est primordial que les plans « bons pour exécution » soient à jour et sans erreur afin d'assurer la constructibilité et la qualité des ouvrages. La liste des statuts d'un artefact (type_statut) est : en cours, terminé, pour exécution, exécuté, à valider, validé, à modifier. Un artefact peut avoir plusieurs statuts (ex. un plan « terminé » et « à valider »).

Conclusion

Cette section présente la méta-modélisation autour du concept de pratique et s'appuie sur les concepts métiers du Méta-Modèle du Contexte Coopératif. La section suivante consiste à détailler la façon dont les acteurs, artefacts et activités sont impliqués autour du concept de pratique individuelle puis d'opération.

### *1.28.2 Les pratiques individuelles et les opérations métiers*

Les pratiques individuelles (PI) peuvent être définies comme le comportement « isolé » de chaque acteur impliqué dans une pratique collective. Elles peuvent être propres à un ou plusieurs rôles mais toujours exécutées de manière indépendante. Celles-ci sont caractérisées par les mêmes attributs que les PC : un « nom » et une « description ». Une des PI qui composent la PC illustrée dans la section précédente peut par exemple être décrite ainsi : « le concepteur partage des plans avec l'ingénieur pour qu'il vérifie la structure ». On peut la nommer plus simplement : « partage des plans ».

Ainsi, le méta-modèle des pratiques individuelles (MMPI, Figure 88) est une évolution du méta-modèle des pratiques collectives (ne figurent plus les différentes énumérations pour faciliter la lecture). Comme on peut le voir, les relations « implique » dans le MMPC sont détaillées par des relations plus spécifiques :

- L'acteur « exécute » une pratique individuelle (ex. la production de plan d'exécution par le concepteur). Une PI peu aussi « concerner » un ou plusieurs autres acteurs (ex. le partage de plans avec l'entreprise).
- Une PI « produit » des artefacts ou « manipule » ceux déjà existants.
- Une PI peut « concerner » une activité particulière (c'est à dire une tâche dans une phase lors d'un projet particulier).

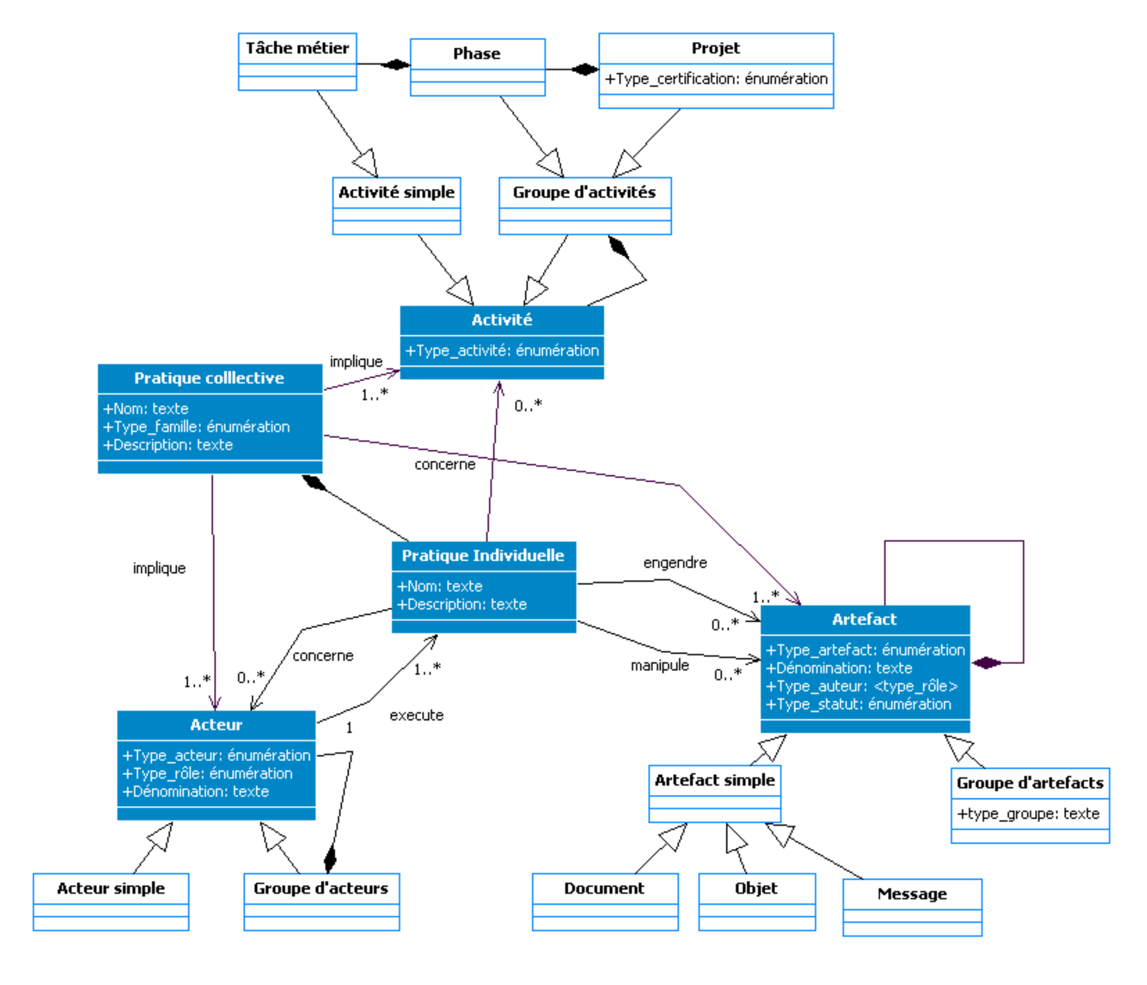



*Figure 88. Méta-modèle de la pratique individuelle (MMPI)*

Caractérisation des opérations : finalisation du MMPM

Dans la théorie de l'activité (présentée précédemment au § 1.1.3), l'opération est le troisième niveau, défini après l'activité et l'action. Les opérations sont les mécanismes qui composent l'action et qui sont réalisés de manière inconsciente. Nous nous servons de ce même concept pour détailler les pratiques individuelles.

Afin de définir ces opérations, nous avons adopté les sous-processus de l'information définis par BO-Christopher Björk (B. C. Björk 1999; B.-C. Björk 2002) et introduit précédemment (au § 1.7.1) : la production, la communication, la diffusion, la récupération. En tant que familles d'opérations, ces quatre sous-processus couvrent l'ensemble des opérations au même titre que les onze familles de pratiques collectives pour les besoins métiers. Le méta-modèle des pratiques métiers (MMPM) est une évolution du MMPI par l'ajout et la caractérisation des opérations métier qui composent une PI.

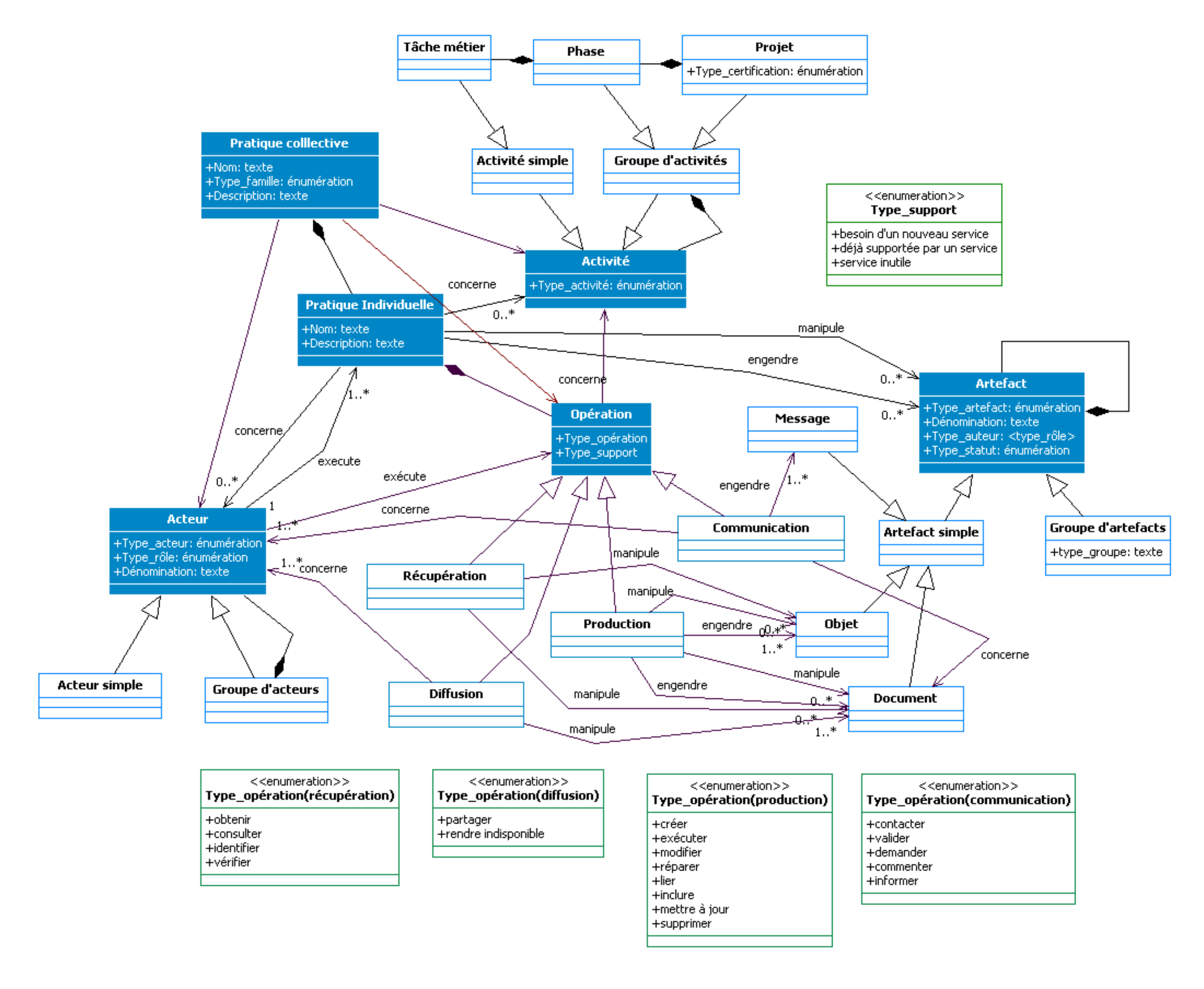



*Figure 89. Méta-modèle complet des pratiques métiers (MMPM)*

À ce niveau de détail, nous avons souhaité manipuler une suite exhaustive d'opérations types pouvant définir un maximum de situations. Les différentes opérations que nous avons identifiées dans chacune des familles sont :

- pour les opérations de communication : contacter, demander, valider, commenter, informer,
- pour les opérations de production : créer, exécuter, modifier, réparer, lier, inclure, mettre à jour, supprimer
- pour les opérations de diffusion : partager et rendre indisponible
- pour les opérations de récupération : obtenir, consulter, identifier, vérifier

Les relations qui lient les opérations avec les éléments du contexte coopératif (c.à.d. les acteurs, artefacts et activités) sont les mêmes que pour les pratiques : engendre, manipule et concerne. Dans le MMO, les couples opérations-éléments sont détaillés et suivent le schéma suivant (Figure 90).

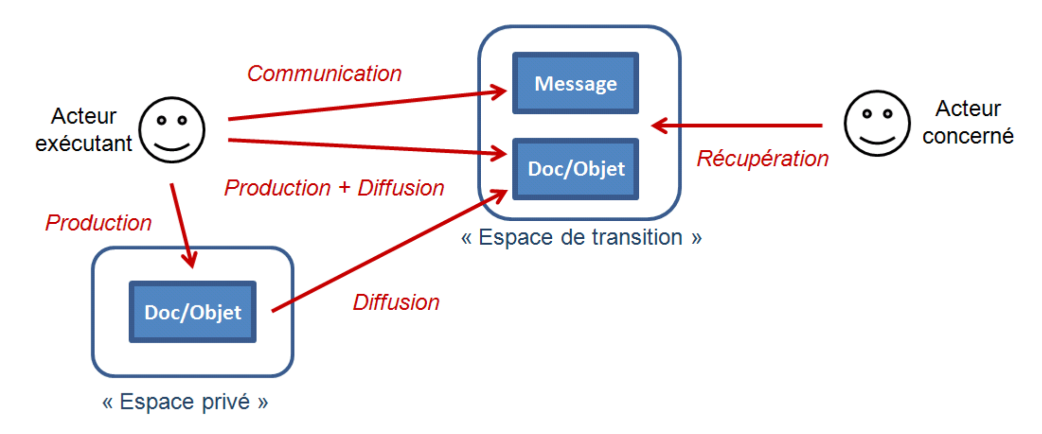



*Figure 90. Les opérations (en rouge) et artefacts manipulés (rectangles bleus) autour des concepts d'espace de transition et d'espace privé*

Les opérations de communication génèrent des messages pour les acteurs concernés. Par exemple, une opération « demander » va générer une requête à laquelle va accéder la personne à qui l'on fait cette requête.

Les opérations de production génèrent (ou manipulent) des documents et des objets. Ceux-ci peuvent dans un premier temps rester privés et/ou être partagés par une opération de diffusion. Ils passent alors par un espace de transition qui sera défini par l'usage (voir chapitre suivant). C'est par exemple le processus habituel suivi pour les plans : ils sont produits par le concepteur puis, une fois « bons pour exécution », ils sont diffusés aux entreprises concernées. Lorsqu'il s'agit de la production d'un ouvrage (exécution), celle-ci se fait la plupart du temps sur le chantier et donc de manière partagée. Par contre, si l'on considère des ouvrages préfabriqués, le processus se rapproche de celui des plans (production puis diffusion dans un deuxième temps).

L'attribut « type-support » servira à identifier si l'opération analysée doit faire l'objet du développement d'un service ou pas. Les « type_support » utilisés sont : « besoin de développement d'un service», « déjà supporté par un service » et « service inutile ». En effet, dans le cadre de l'adaptation de services, il ne s'agira pas de tout reconcevoir mais au contraire de tenir compte des solutions existantes. Savoir quand redévelopper une solution ou au contraire quand réutiliser une solution existante est le premier enjeu afin éviter d'engager des coûts inutiles. Cela permet ainsi de diminuer les risques d'échec du projet, et ceci dès ses premières étapes.

### *1.28.3 Conclusion*

Le Méta-Modèle du Contexte Coopératif (MMCC), issu de travaux antérieurs, a été réutilisé et transformé pour créer le Méta-Modèle des Pratiques Métier (MMPM). Outre les concepts métiers du MMCC, ce dernier a été créé sur la base de trois autres concepts centraux : les pratiques collectives, les pratiques individuelles et les opérations métier.

Ce MMPM doit nous permettre de décrire les pratiques observées dans des projets de conception-construction collaborative grâce aux concepts qu'il définit, à leurs attributs et aux relations qui les relient. Il s'agit en d'autres termes, d'instancier ce méta-modèle unique en autant de modèles de pratiques que de d'analyses effectuées.

Pour cela, il s'agit d'utiliser des formalismes adaptés et les outils d'édition qui conviennent. Tel que cela a été introduit en 1.24.1, nous avons utilisé le framework GMF de l'outil de développement Eclipse pour créer notre propre éditeur de modèles.

## 1.29 Le modèle de pratiques

Le modèle de pratiques est une représentation conceptuelle de pratiques observées au travers d'une analyse métier. Il utilise les concepts du MMPM qu'il représente dans un formalisme particulier. Nous proposons pour cela notre propre formalisme et notre propre éditeur.

### *1.29.1 Critique d'un formalisme existant*

La Figure 91 issue de travaux préliminaires (Zignale et al. 2011) illustre un premier essai de modélisation d'une pratique collective (PC) que nous avons réalisée, celle de l'évaluation des documents de conception par un expert. Dans cette PC, le concepteur fournit les documents de conception au maître d'ouvrage qui les soumet à un expert pour évaluation. Souvent, le maître d'ouvrage délègue sa pratique à un assistant à la maîtrise d'ouvrage qui possèdera davantage de compétences pour cela.

Le formalisme qui a été utilisé ici est le diagramme d'activités UML. PC, PI et opérations sont des actions que l'on distingue graphiquement. La PC « couvre » tous les acteurs alors que chaque Pratique Individuelle (PI) qui la compose se situe dans la « swimlane » d'un acteur en particulier. Il en va de même pour les opérations. Ce diagramme illustre également les flux d'artefacts engendrés ou simplement manipulés par les opérations.

Suite à cette expérimentation, nous constatons que le diagramme d'activités n'instancie qu'une partie notre méta-modèle de pratiques : tous les attributs de chaque classe ne sont pas traités (ex. les types d'artefacts, les familles de pratiques…), et les types énumérés ne sont pas utilisés. De plus, la distinction entre certains éléments doit se faire manuellement. Par exemple, les PC, les PI et les opérations sont ici toutes représentées par des « actions » en langage UML. Il est nécessaire, pour les distinguer lors de leur modélisation, de spécifier leur stéréotype et de modifier leur aspect graphique. Il en va de même pour les artefacts qui sont tous indistinctement des « objets ». Le méta-modèle du diagramme d'activités et le MMPM étant différents, le diagramme d'activités ne permet donc pas une bonne instanciation du MMPM.

La modélisation des pratiques de chaque acteur dans un même modèle en alourdit la lecture et nous parait superflue. Le séquençage des pratiques (flèches rouges dans le modèle)n'est d'ailleurs pas décrit par notre méta-modèle, le but étant en premier lieu de comprendre ce que chacun fait et non d'optimiser un processus qui devra être suivi. Il ne doit donc pas être modélisé. Par contre, répartir la modélisation de chaque PI dans des diagrammes indépendants permettra alors d'isoler les besoins de chacun et en facilitera la lecture.

Nous pensons qu'un formalisme de modélisation doit répondre aux questions que le concepteur se pose lors de la modélisation. Dans notre cas, il est question de créer un formalisme qui soit utilisable et compréhensible par les analystes métiers et donc à partir des concepts métiers définis par le MMPM. De plus, nous préconisons un formalisme simple d'accès, autant par la lecture que par l'édition.

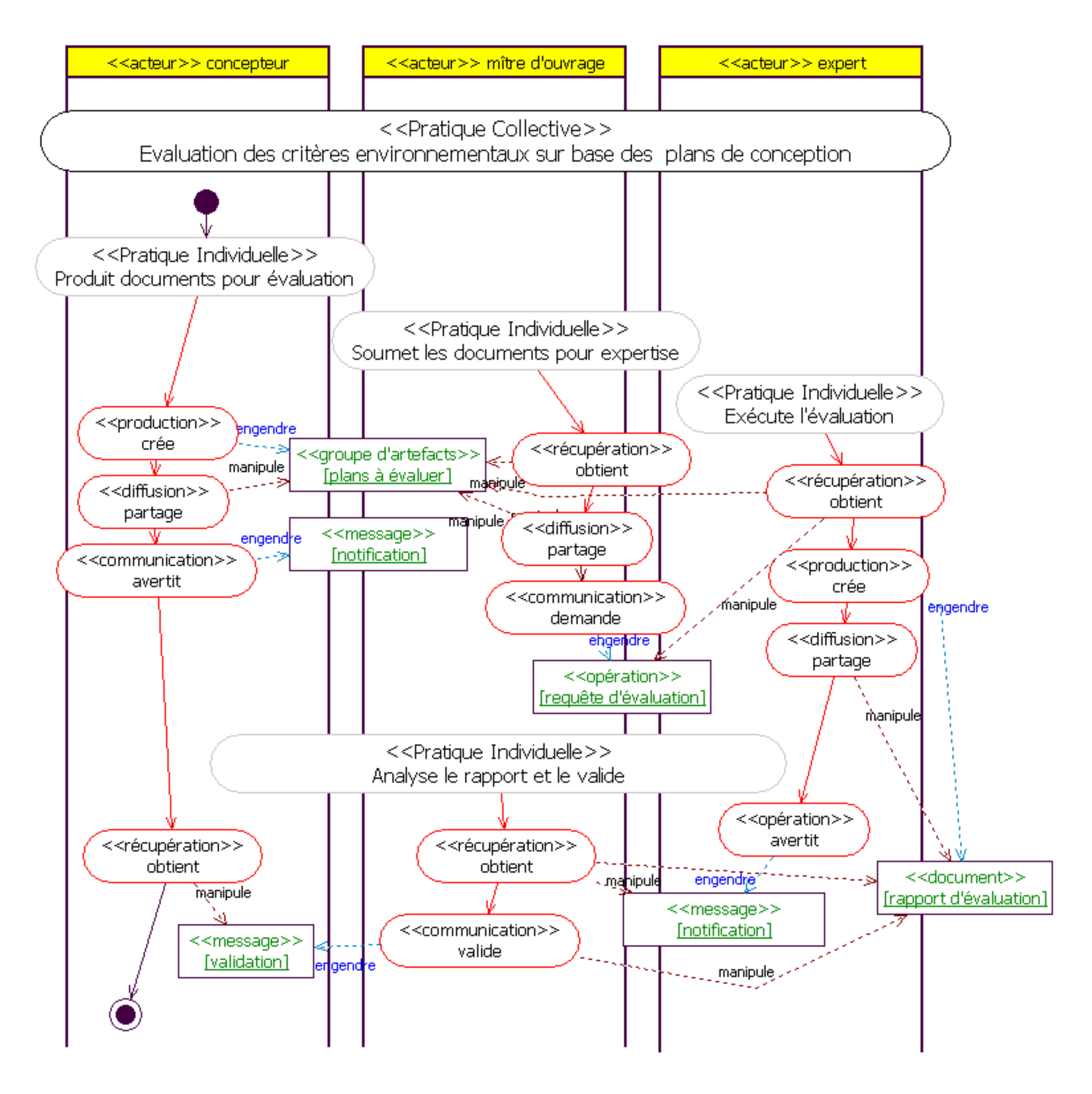



*Figure 91. Instanciation du MMPM par un diagramme d'activités UML modifié*

<u>Vers une modélisation par des arbres hiérarchiques</u>

Il nous paraît nécessaire d'introduire un nouveau formalisme pour instancier notre MMPM plutôt que de réutiliser le diagramme d'activités. L'objectif est de pouvoir manipuler les concepts introduits afin de guider la modélisation, en utilisant une palette d'outils adaptée et les différentes énumérations du méta-modèle.

Nous avons utilisé le Framework GMF d'Eclipse (présenté en 1.24.1) afin de créer un éditeur permettant d'instancier notre méta-modèle de pratiques sous forme d'un arbre hiérarchique. Ce mode de visualisation est utile pour représenter des systèmes relationnels organisés en couches comme par exemple la structure d'une entreprise ou d'une famille (cf. arbre généalogique). Nous verrons qu'il s'avère plutôt adapté pour instancier le méta-modèle de pratiques.

Afin d'en faciliter la diffusion et l'évaluation dans la communauté scientifique, cet éditeur a été conçu en anglais.

### 1.29.2 *L'éditeur et les modèles générés*

La Figure 92 illustre un exemple de modèle de pratique, tel que nous l'avons conçu, accompagné de la palette de l'éditeur utilisé pour le décrire. Le cas choisi est le même que précédemment : il concerne une pratique collective d'échange de plans pour leur expertise. Le modèle se focalise cependant sur la PI du concepteur. Cela permet, comme énoncé précédemment, de générer un modèle relativement simple et facile à lire rapidement. Comme nous pouvons le voir, ce modèle représente couche après couche :

- la **PC** considérée, définie par son nom, sa famille, une description ;
- l'**acteur** type auquel on s'intéresse, décrit par son rôle, sa dénomination, son métier ainsi que la **PI** de cet acteur, définie par un nom et une description ;
- les **opérations**, pour lesquelles on renseigne le type et le besoin de service
- et enfin un **artefact** (défini par un type, une dénomination, un auteur, un état) manipulé ainsi que le **groupe d'acteurs** concerné (nature du groupe, rôle, spécificité).

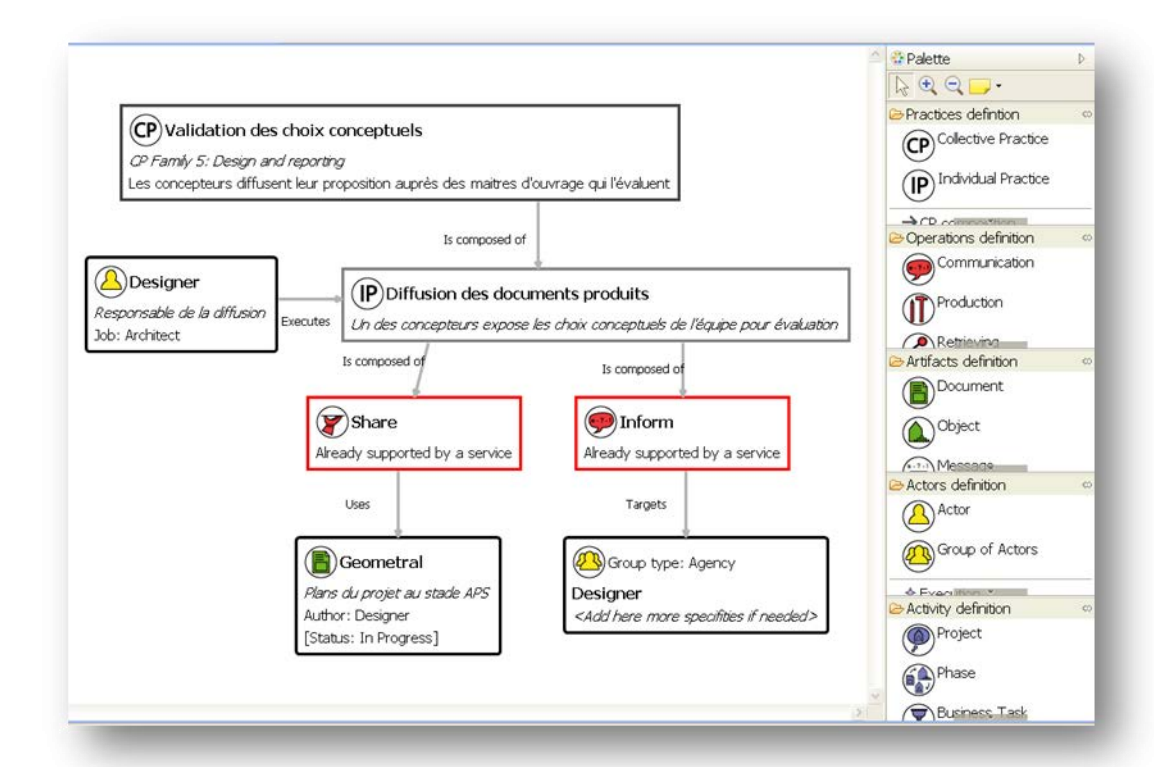



*Figure 92. Interface de l'éditeur GMF d'arbres hiérarchiques pour la modélisation des pratiques (exemple)*

Chacun de ces éléments est introduit dans le modèle à partir de la palette (à droite). Ces éléments sont donc des concepts du méta-modèle proposé et permettent d'homogénéiser les analyses. Chaque élément est ensuite éditable, notamment par le choix d'attributs dans des énumérations prédéfinies. L'expression littérale permet de faire des descriptions qui vont au-delà de ces listes et qui permettent notamment d'apporter de l'information supplémentaire. C'est notamment ce type d'édition que nous utilisons pour les attributs « nom » ou « description ». La combinaison des deux modes d'édition offre une description à la fois généralisable et spécifique.

L'avantage de l'éditeur développé est de pouvoir formaliser de manière structurée une analyse métier en créant un modèle adapté. Nous tirons pour cela profit de la caractérisation du contexte coopératif et de celle des pratiques au travers de méta-modèles. L'objectif est de permettre une modélisation qui soit cohérente avec le domaine analysé, à savoir le projet de conception/construction architecturale.

Ainsi, à partir de chaque classe du méta-modèle de pratiques, un élément de diagramme est créé et peut être ajouté dans le diagramme via la palette. On instancie ainsi chaque classe puis ses attributs, de la manière qui a été définie dans le méta-modèle, à savoir par choix dans une liste (attributs de type énumération) ou par saisie de texte (attributs de type texte). Un attribut peut être instancié plusieurs fois si cela a été spécifié dans le méta-modèle grâce aux cardinalités. La création de relation entre éléments est contrainte par les relations du méta-modèle, et également par leurs cardinalités. Par exemple, la cardinalité 0.1 sur la relation « exécute » entre un acteur et une pratique individuelle implique qu'un seul élément « acteur » pourra être relié à chaque élément pratique individuelle dans le modèle. Cela permet de respecter la définition d'une pratique individuelle, qui est « le comportement isolé de chaque acteur impliqué dans une pratique collective » (voir 1.28.2).

Le framework GMF (présenté en 1.24.2) propose des fonctionnalités qui assistent la génération de l'éditeur à partir du méta-modèle, notamment grâce à un tableau de bord (Figure 93). Ces fonctionnalités nous ont permis d'entreprendre la conception de cet éditeur sans connaissances en matière de développement et à travers un apprentissage rapide. Le bémol de ces fonctionnalités est qu'une fois l'éditeur généré et personnalisé (couleurs, formes des éléments, types de traits…), toute régénération automatique aura pour effet d'effacer la personnalisation. Il est donc important de définir le méta-modèle de manière très précise avant de commencer la spécification d'un éditeur qui sera pertinent.

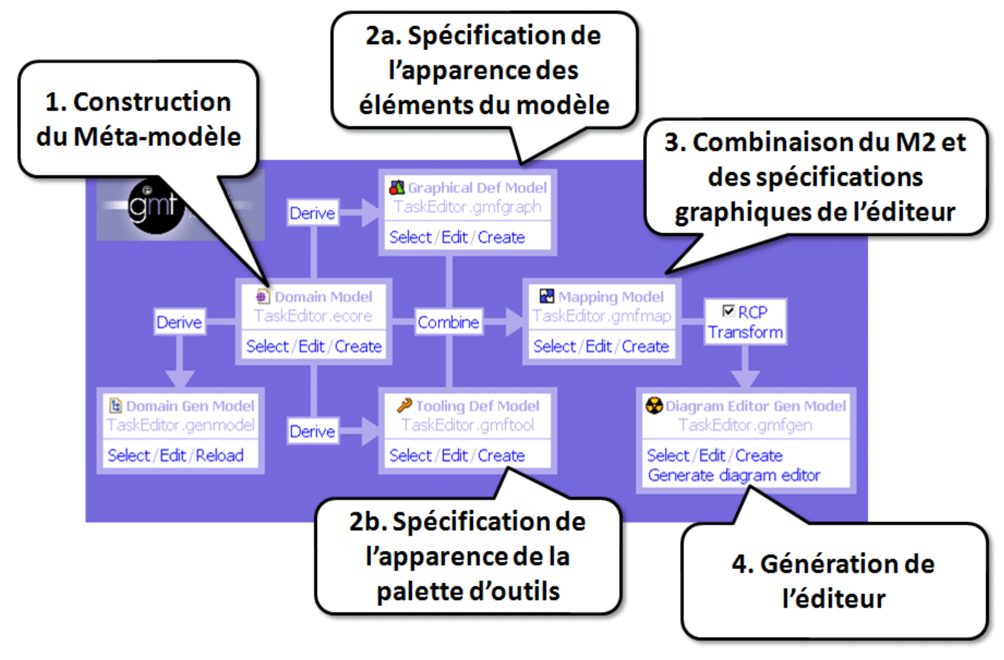



*Figure 93. Le tableau de bord GMF*

Bien évidemment, il est toujours possible de regénérer « manuellement » et par parties. Après un apprentissage plus approfondi, il nous est possible de modifier sensiblement le méta-modèle et de répercuter ces changements sur l'éditeur. Cela inclut, du plus simple à mettre en œuvre au plus compliqué :

- l'ajout de choix dans une énumération ou le changement d'une cardinalité,
- l'ajout d'un attribut à une classe,
- l'ajout d'une classe et/ou d'une relation.

Cette possibilité d'adapter l'éditeur est très importante vis-à-vis des perspectives expérimentales de notre travail. En effet, au fil d'applications futures de notre méthode, il est possible que les concepts évoluent voire que de nouveaux soient introduits au fil des analyses métier qui seront menées. Cela a déjà été le cas au travers des trois expérimentations que nous avons menées (voir chapitre 12). L'exemple le plus typique est l'ajout dans une énumération d'un élément qui n'aurait pas été spécifié dans un premier temps mais qui est rencontré dans plusieurs cas. Ainsi, les allers-retours entre modélisation et méta-modélisation assureront une analyse métier toujours plus pertinente.

### *1.29.3 Intégration dans le cahier d'exigences*

Le cahier d'exigences a été introduit comme un moyen de rassembler les modèles de chaque point de vue (organisationnel, opérationnel et fonctionnel). Le point de vue organisationnel, formalisé au travers des diagrammes de pratiques, compose donc la première partie de ce document.

Comme nous venons de le remarquer, la justesse des modèles pourra évoluer au fil des analyses, à travers l'amélioration du méta-modèle. Pourtant, il est important de pouvoir faire des analyses pertinentes, même si la maturité des modèles ne le permet encore pas. C'est pourquoi, comme nous l'avons introduit dans le paragraphe présentant le cahier d'exigences (§ 1.25.2), celui-ci comprend des champs d'édition libre qui permettent de pallier les manques du méta-modèle.

De manière générale, les champs libres du cahier d'exigences seront un support d'expression pour les acteurs du projet de développement, quelque soit leur point de vue. Nous partons du postulat que toute idée ou remarque au cours du processus peut avoir un impact sur la pertinence du résultat, à savoir l'adaptation du service à développer. Le cahier d'exigence supporte ainsi la formalisation de ces remarques et assure leur capitalisation et leur prise en compte par les différents participants potentiellement concernés. La représentation du modèle et de ces informations additionnelles prend la forme d'un tableau (Figure 94).

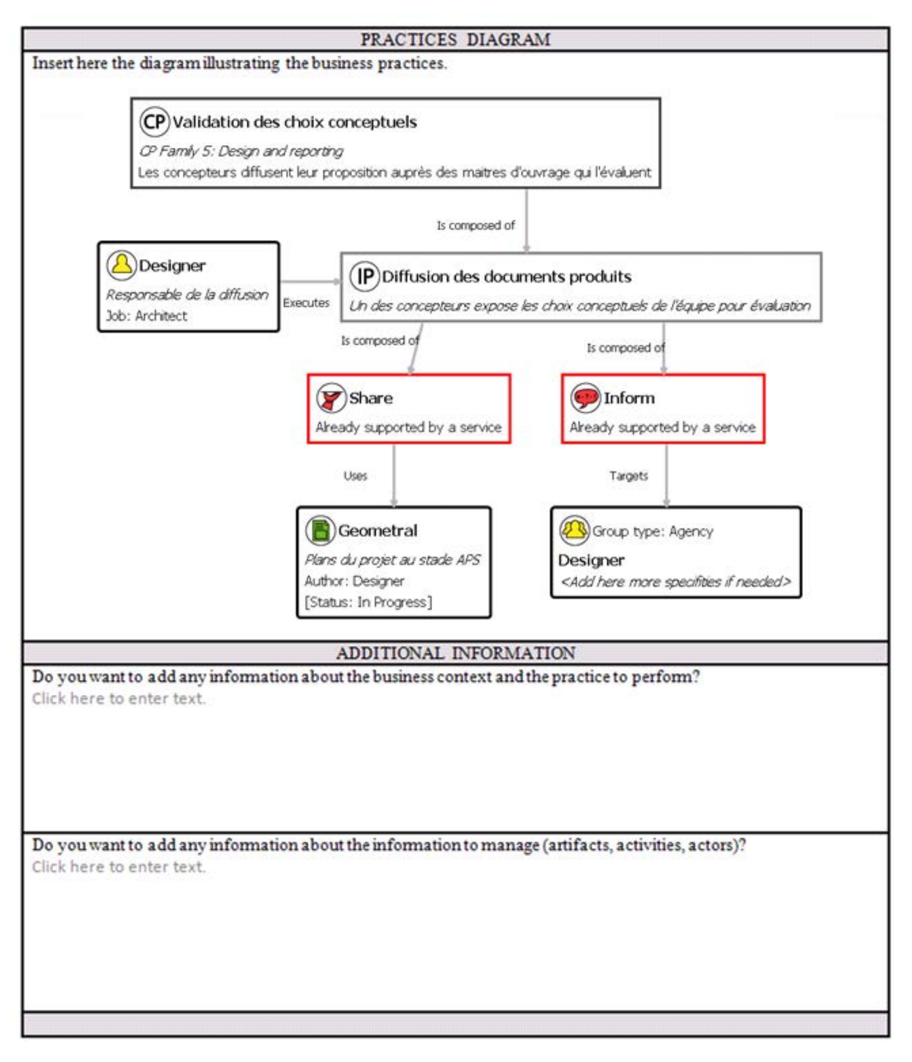



*Figure 94. Formalisation de l'analyse métier, combinaison d'un modèle et de remarques*

## 1.30 Conclusion

Les études sur le contexte coopératif, la théorie de l'activité ou encore les processus de l'information (c.f. § 1.1 puis 1.28.2) nous ont permis de conceptualiser nos propres analyses portées sur le déroulement de la coordination au cours d'un projet AIC. Nous avons construit un méta-modèle des pratiques de projet autour de trois concepts : les pratiques collectives, les pratiques individuelles et les opérations. Nous avons ensuite développé un éditeur pour instancier ce méta-modèle afin de décrire des pratiques observées.

Les pratiques collectives analysées dans ce chapitre ont été réparties en onze familles qui apportent une structuration du projet plus fine que les traditionnelles phases. Elles définissent des objectifs généraux relatifs à des groupes de projet impliqué dans des tâches collaboratives. Les pratiques individuelles raffinent ces objectifs du point de vue de chaque acteur considéré individuellement. Elles sont davantage précisées par des opérations réparties en quatre familles et dont l'exécution a pour but de remplir l'objectif identifié pour chaque acteur.

Les pratiques et opérations décrivent finement le déroulement d'une activité métier, mais elles sont pensées indépendamment des outils qui seront utilisés pour effectuer cette activité (que ce soient des outils matériels ou logiciels). Notre postulat est que cet outillage doit être défini à posteriori.

L'outillage consiste à médiatiser chaque opération, acteur, artefact ainsi que les espaces (privé et de transition) qui leur sont associés (rappel : voir Figure 90). Les études sur le TCAO et notamment les dimensions fonctionnelles (distance et temporalité) nous permettront de catégoriser ces moyens de médiatisation. Les trois scénarios illustrés dans la figure suivante (Figure 95) montrent comment une même pratique métier peut être médiatisée différemment : nous parlerons alors d'usages différents. L'espace de transition y prend alors plusieurs formes et dimensions. Par exemple, dans le premier cas (usage 1), il s'agit de « l'entre-deux » dans le très court instant pendant lequel le document passe de la main d'un acteur à celle d'un autre. Dans le deuxième cas (usage 2), c'est un service de courrier comme la « Poste » qui assure cette transition. Enfin, le troisième cas introduit l'usage de services informatiques qui médiatisent le partage d'information sous forme numérique (usage 3). C'est à ce type d'usage que nous nous intéressons particulièrement dans la suite de ce manuscrit, en cherchant à le caractériser de la même manière que nous venons de le faire avec les pratiques.




*Figure 95. Des usages différents pour une même pratique, exemple d'une pratique de « partage de documents »*

# Chapitre 10 – La modélisation des usages : le point de vue opérationnel

Ce chapitre introduit le concept d'usage que nous avons adopté pour caractériser la médiatisation des pratiques métiers par des outils numériques. Concevoir un usage consiste à considérer l'acteur d'un projet comme utilisateur d'un outil puis à définir l'interaction entre les deux, la manière dont est traitée l'information métier numérisée ainsi que le contexte de l'usage. La construction d'un méta-modèle d'usage nous permet de caractériser ces concepts, et les formalismes utilisés pour les décrire sont issus des domaines du Génie Logiciel et de la conception d'IHM. Nous verrons comment nous adaptons ces formalismes pour lier la description d'un usage à celle de la pratique qu'il médiatise.

## 1.31 Définition et concepts

### *1.31.1 Usage et utilisation*

Au travers du point de vue opérationnel, on cherchera à exprimer la manière dont est « médiatisée » une pratique métier par l'introduction d'un outil numérique. Médiatiser signifie littéralement « avoir une fonction d'intermédiaire ». Nous définirons dans quelle mesure l'outil assure cette fonction d'intermédiaire à travers le concept d'usage.

Les termes usage et utilisation sont deux synonymes qui peuvent tous deux signifier l'emploi de quelque chose. On confère cependant à l'usage plus d'acceptions[32]. Il est par exemple relatif à une habitude (ex. l'usage du chapeau) ou à des règles établies (ex. il est d'usage de mettre une cravate ici). Ainsi, le terme usage définit une utilisation commune, normale, prévue alors que l'utilisation se rapporte plutôt à un point particulier, une situation donnée. Par exemple, on dit que l'usage d'une chaine hifi est de lire de la musique mais que l'utilisation de celle-ci en extérieur n'est pas recommandée. Par extension, l'usage se réfère à l'appropriation : on peut par exemple

[32] http://french.stackexchange.com/questions/722/usages-dusage-et-dutilisation-quel-mot-utiliser

parler d'usage détourné lorsqu'il s'agit d'un outil qui sert à autre chose que ce à quoi il a été destiné.

Pour caractériser la médiatisation d'une pratique métier par l'emploi d'un outil, nous privilégions donc le terme d'usage plutôt qu'utilisation car il couvre une définition plus large que celle du simple emploi. Nous cherchons en effet à retrouver, en plus de la notion d'emploi, les notions :

- d'habitude, qui découle du concept de pratique que l'on veut outiller, qui est par nature une manière de travailler récurrente
- d'objectif, relatif à l'objectif métier, le but étant de se servir des outils pour les atteindre,
- et d'appropriation par les acteurs, notamment en fonction de leur rôle duquel découlent leurs besoins.

En gardant à l'idée ces acceptions des termes usage et utilisation, on remarque également que l'évolution de l'usage peut-être dépendante de celle de l'utilisation. On dira par exemple que l'usage des plateformes collaboratives ou encore des applications mobiles a évolué dans le domaine AIC, car leur utilisation est devenue plus intuitive. On retrouve ici la notion de « push technologique » introduite précédemment. Les avancées technologiques permettent en effet de supporter une charge de travail toujours plus soutenue et contribuent ainsi à l'évolution des usages.

Si du point de vue de la terminologie notre choix se tourne donc vers l'adoption du concept d'usage, la littérature relative à la conception d'HM valide également ce choix. D'après la description qui a été faite au chapitre 4 (au § 1.10.1, Tableau 4) les enjeux de l'approche de conception centrée-usage de (L. L. Constantine & Lockwood 2003) semblent en effet correspondre aux nôtres :

- la concentration sur l'outillage des pratiques métiers,
- une approche dirigée par les modèles,
- une implication sélective de l'utilisateur,
- des processus systématiques et entièrement spécifiés,
- une adaptation assurée par l'ingénierie et non par la résolution itérative d'erreurs.

### *1.31.2 Caractérisation de l'usage*

L'usage est donc relatif à un objectif qui le motive. Cet objectif est dérivé de la pratique métier qu'il médiatise : on détaillera par l'usage la « manière de » réaliser cette pratique. Nous chercherons à définir les usages pour chaque acteur, ainsi c'est à la pratique individuelle que l'on fait référence ici. Par exemple, pour la pratique « diffuser les documents », l'usage pourra consister à « diffuser les documents par mail ». En d'autres termes, on pourra dire qu'il s'agit de « faire l'usage du mail pour diffuser les documents ».

Tel que nous l'avons vu dans le chapitre 4, la conception d'IHM « enrichit » la conception logicielle par l'analyse de trois éléments principaux : les tâches, le contenu et le contexte. Nous nous servirons de ces trois concepts pour caractériser l'usage, à la lumière des différentes études de l'état de l'art analysé.

Les tâches définissent l'interaction entre l'utilisateur et l'outil. Elles détaillent cette médiatisation en décrivant le comportement de l'utilisateur et de l'outil lors de l'usage. Nous utiliserons plusieurs niveaux d'abstraction pour définir ces tâches.

L'information manipulée lors de l'exécution des pratiques est aussi médiatisée par l'outil à travers son usage. C'est ce que nous appelons le contenu d'interaction. Il est composé d'objets (les objets d'interaction) qui sont définis par une donnée (ce qui est manipulé) et une forme (comment il est représenté).

Le contexte de l'activité collective et le contexte de l'acteur, deux des trois parties du contexte tri-facettes (cf. § 1.2.3), sont caractérisés par le Méta-Modèle des Pratiques Métier au travers des concepts de Pratique Collective et Individuelle. Il s'agit à présent de caractériser la troisième partie qui correspond au contexte utilisateur. Conformément à notre terminologie, nous l'appellerons ici le contexte d'usage.

## 1.32 Le méta-modèle des Usages

À partir des définitions que nous venons d'introduire, nous pouvons définir les deux éléments de caractérisation d'un usage (Figure 96) :

- l'objectif à atteindre en tant qu'attribut de l'usage,
- la relation médiatise qui relie conceptuellement un usage à une pratique individuelle.

Nous pouvons dès à présent spécifier (au travers des cardinalités) que :

- plusieurs usages peuvent médiatiser la même pratique (par exemple si il y a changement d'outil, de lieu…)
- un même usage peut répondre à plusieurs pratiques (si le déroulement des ces pratiques est similaire).

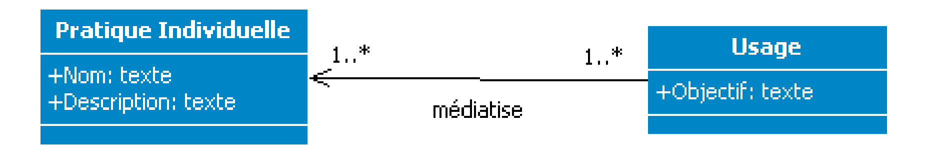



*Figure 96. Relation entre les concepts d'usage et de pratique*

Nous composons le Méta-Modèle d'Usage (MMU) par parties, comme nous l'avons fait pour caractériser les pratiques. Les sections suivantes montrent comment cette relation « médiatise » entre usage et pratique se répercute sur les éléments relatifs aux deux concepts :

- entre les tâches et les opérations métier,
- entre le contenu d'interaction et les informations manipulées par ces opérations,
- entre l'utilisateur (dans son contexte) et l'acteur.

### *1.32.1 L'interaction*

Nous décomposons l'interaction de l'utilisateur avec le système en deux phases : l'action cognitive nommée « l'intention utilisateur », et l'action physique qui elle est appelée la « tâche utilisateur ». La responsabilité de l'outil est appelée « tâche outil ».

L’intention relève d’un objectif général (ex. s’identifier). Certaines des intentions sont déduites des opérations métiers qui ont été définies dans le méta-modèle des pratiques. Cela est formalisé par la relation « supporte ». D’autres intentions sont indépendantes des opérations métiers et relève plus d’une contrainte technique. Par exemple, « s’identifier » qui est nécessaire pour beaucoup d’applications mais que l’on n’associe pas à une opération métier particulière.

Les tâches utilisateurs sont dans un premier temps considérées comme abstraites. C'est-à-dire qu’elles décrivent une action précise (ex. entrer login) et non comment cette action est exécutée (ex. taper login, dire le login…). Cela relève de la tâche dite concrète qui peut être de l’ordre de l’interaction ou de la perception seulement.

Les méthodes de conception d’IHM et plus particulièrement dans les approches génératives de MDA (voir § 1.11.2) introduisent des types de tâches abstraites que nous réutilisons. Cela nous permet d’envisager des perspectives d’intégration de notre méthode dans de telles approches. Ces types sont : entrée, sortie, commande, sélection, navigation, conteneur.

De la même manière nous typons les tâches concrètes :

- d’interaction : écriture, choix, déclenchement, manipulation, transformation, défilement, autre…
- de perception : graphique, audio, matérielle.

Trois relations caractérisent un processus de tâches. Cette dynamique était absente dans la caractérisation des pratiques métiers. Nous partions en effet du postulat que l’ordre des opérations métiers n’importe pas car il peut varier d’une situation à une autre et selon la manière dont ces opérations seront outillées. Cependant, dans la caractérisation de l’usage et des tâches qui le composent, il est nécessaire d’intégrer cette notion afin de composer la relation entre les interfaces développées. Les relations identifiées sont :

- La relation « OU » spécifie que l’on exécute une des tâches décrites
- La relation « ET » spécifie que toutes les tâches décrites doivent être exécutées
- La relation « PUIS » ajoute un ordre dans l’exécution des tâches.

Nous retrouvons ici les concepts manipulés par CTT (Paternò et al. 1997) ou K-MAD (Baron et al. 2006), les langages de modélisation de tâches d’interaction, en même temps que les concepts d’intention de l’utilisateur et de responsabilité du système dans les « essential use case » de la conception centrée-usages (L Constantine 2001) (cf. § 1.10.3). Le méta-modèle suivant (Figure 97) représente ces concepts et leurs relations.

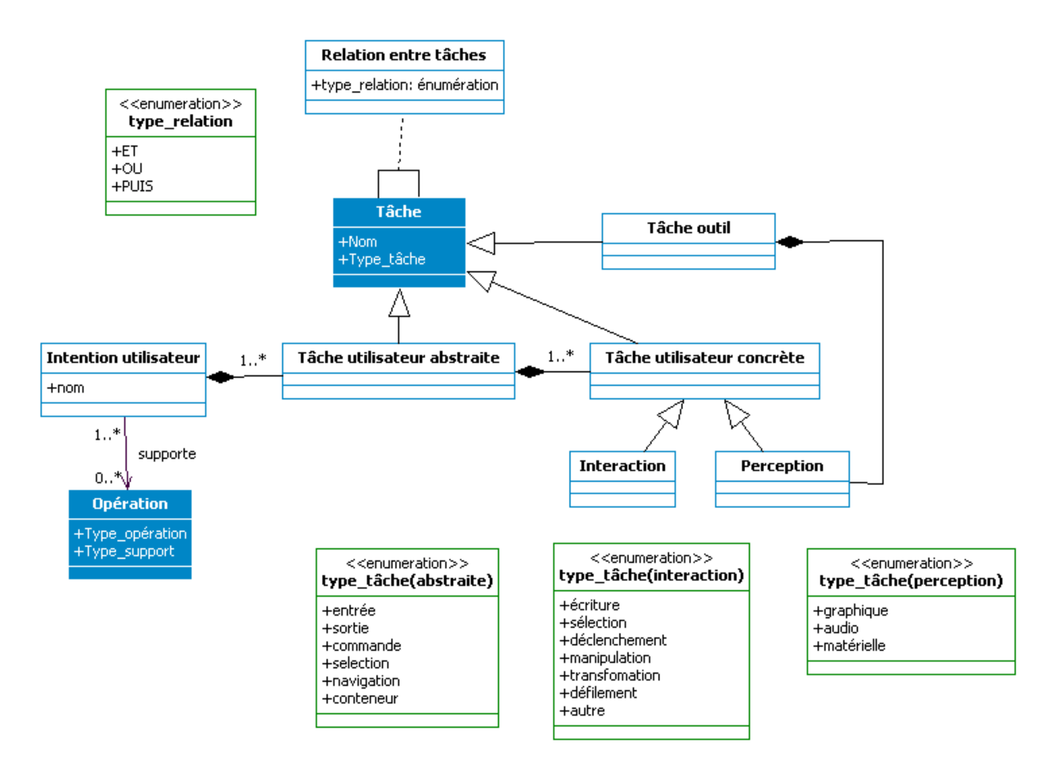



*Figure 97. Caractérisation des intentions et des tâches*

### *1.32.2 Le contenu d'interaction*

Le contenu d'interaction est composé d'objets d'interaction (OI) qui représentent les éléments du contexte coopératif manipulés au cours d'une pratique et médiatisés par un usage. Ce sont les éléments avec lesquels l'utilisateur devra interagir lors de l'exécution d'une tâche.

Un OI est la combinaison d'une donnée et d'une représentation. Il a un niveau de description abstrait, mais devra contenir assez d'information pour en faire un objet concret lors du développement du système. À partir de la littérature dans le domaine des IHM, nous nous sommes inspirés des objets d'interaction de la méthode Symphony (S Dupuy-Chessa 2011) mais aussi du concept d'objet d'interaction abstrait décrit dans (JM Vanderdonckt & F Bodart 1993) ou illustré dans (LL Constantine 2003) (voir 1.10.3). Afin de pouvoir définir des objets d'interaction pertinents par rapport au domaine AIC, nous avons également considéré les objets IFC dans le BIM (J. Zhang et al. 2012) et dans les méthodes de conception basées sur l'IDM (CM Eastman & Jeong 2010).

Le nom d'un OI est relatif à la donnée qu'il représente. Cette donnée, pouvant être définie par plusieurs attributs, peut être un élément du contexte coopératif comme :

- un artefact, que ce soit un document (ex. un rapport médiatisé par un fichier texte), un objet (ex. un ouvrage médiatisé par un dessin numérique) ou un message (ex. une réaction médiatisée par un commentaire sur un fichier)
- un acteur (sous la forme d'un contact dans un annuaire par exemple),
- ou une activité (comme une tâche que l'on manipule dans un diagramme de gantt).

Il peut s'agir également d'une donnée dite primitive (comme par exemple le nom d'un document, la durée d'une tâche dans un planning, l'identifiant d'un acteur…). Une donnée primitive peut être l'attribut d'une autre donnée mais ne possède pas d'attributs elle-même.

Les types de données retenus pour la caractérisation des OI sont :

- (pour les données primitives) booléenne, numérique, date, texte et hypertexte
- (pour les autres données) le fichier, le média et « l'ouvrage numérique » (« building information ») qui est un type de donnée propre au métier et qui nous permettra de typer plus précisément les informations traitées.

Parmi les formes que prennent les objets, on retrouve les objets d'interface communs comme les icônes, les fenêtres, les curseurs, les boutons… On trouve ce type d'objet dans toute interface graphique. D'autres formes sont relatives au domaine lié et notamment à la nature des échanges dans le métier considéré comme par exemple les éléments de dessin (composés de points, de lignes, de surfaces, de formes, d'espaces...). Vu les multiples moyens possibles pour définir ces éléments de dessin, l'attribut « forme » est éditable par un texte libre.

Les couples donnée/forme sont variés et permettent de traiter de multiples cas de figure dans la spécification de l'information médiatisée avec laquelle l'utilisateur interagit. Le Tableau 23 illustre trois exemples de données définies conformément à la caractérisation que nous venons de faire : par leur nom, leur type, leurs attributs (si non primitive) et dans le cas présent deux formes (ou représentations) possibles.

Tableau 23. Exemples de données et d'association de formes

| Donnée manipulée | Type de donnée | Attributs possibles | Formes possibles |
|---|---|---|---|
| Document | Fichier | Nom, type fichier, taille, date création… | Ligne dans un tableau |
| | | | Icône + texte |
| Mur | Building Information | Longueur, largeur, composition, coefficient thermique… | Dessin 2D (surface) |
| | | | Dessin 3D (forme) |
| Quantité de béton dans un mur | Valeur numérique | Donnée primitive (pas d'attributs) | Curseur |
| | | | Type de hachure dans un dessin |

Le diagramme de classes UML suivant (Figure 98) représente la partie du méta-modèle d'usage caractérisant les objets d'interaction. Les relations « représente » entre l'objet d'interaction et chaque élément du contexte coopératif caractérisent à la fois la représentation de ces éléments en tant que données « complètes » ou en en tant que données primitives.

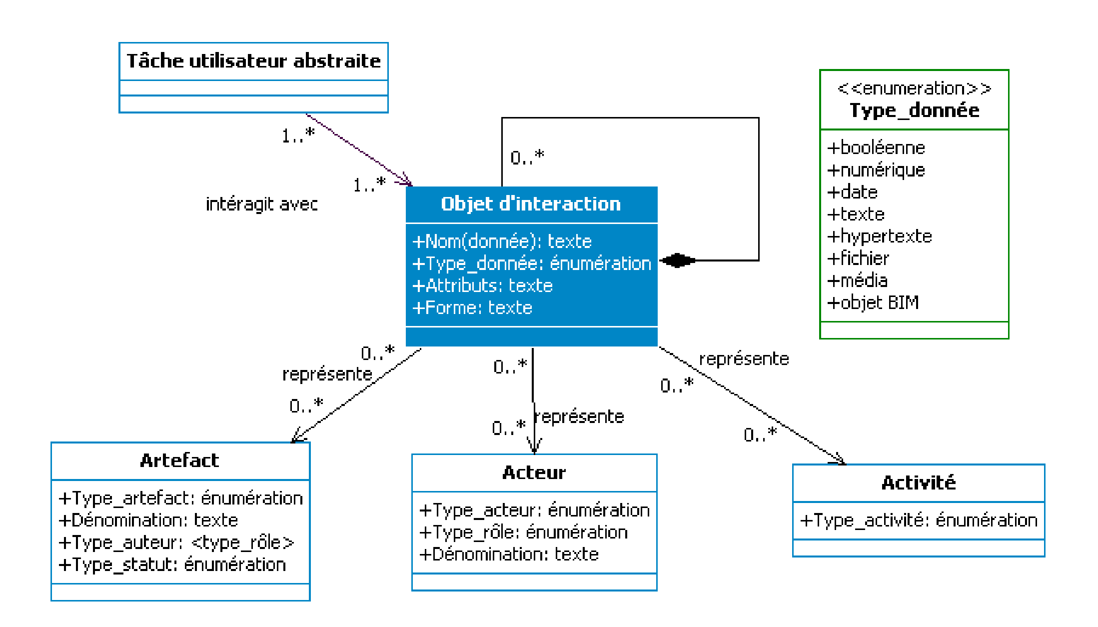



*Figure 98. Caractérisation de l'objet d'interaction*

### *1.32.3 Le contexte d'usage*

Définir le contexte dit « contexte d'usage » n'est autre que considérer un acteur du projet en tant qu'utilisateur d'un outil particulier. Il convient alors de les définir tous deux ainsi que la temporalité et la localisation de l'usage (Figure 99). Les liens conceptuels créés à cet effet sont :

- la relation « représente » entre l'utilisateur et l'acteur,
- la relation « utilise » entre l'utilisateur et l'outil,
- la relation « se déroule » entre l'usage et la temporalité ainsi qu'entre l'usage et la localisation

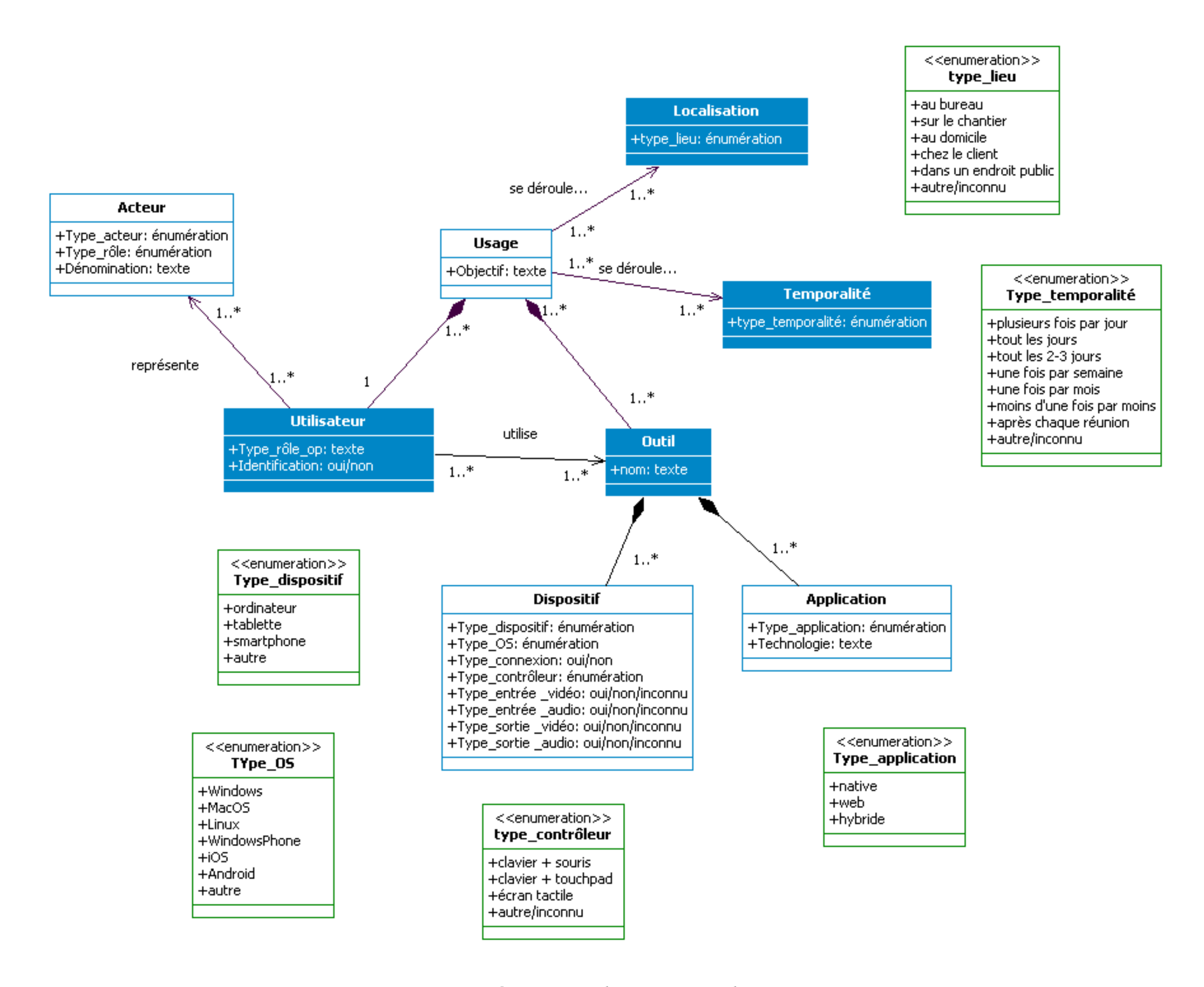



*Figure 99. Caractérisation du contexte d'usage*

On attribue à un utilisateur un rôle opérationnel (type_rôle_op), qui déterminera l'accès ou non à certains services. Ce rôle opérationnel est ainsi dépendant du type de service visé dans le projet de développement. Il peut être attribué en fonction du rôle organisationnel qui est défini dans le Méta-Modèle des Pratiques Métier (MMPM). En d'autres termes, cela implique qu'un acteur respecte les usages qui lui auront été attribués. Par exemple, le rôle opérationnel de « rédacteur » d'un compte-rendu de réunion chantier est en général attribué au coordinateur du chantier (rôle organisationnel). Cette assignation de rôle permet, lors de l'usage d'un outil comme CRTI-weB, de rendre disponibles aux personnes concernées les services de rédaction de compte-rendu en ligne proposés (service « comptes-rendus »). Concernant le service « documents », les rôles organisationnels sont « utilisateur », « administrateur projet » et « superviseur projet ». Le choix actuellement limité aux rôles opérationnels de CRTI-weB pourra être étendu, notamment si l'on considère les outils BIM (ex. BIM manager). Actuellement, nous ne proposons donc pas d'énumération des types de rôle opérationnels possibles mais le méta-modèle pourra évoluer.

Par contre, nous jugeons important de spécifier si l'utilisateur aura besoin de s'identifier (identification : oui/non) pour accéder à l'usage considéré (le mode d'identification le plus

fréquent est l'attribution d'un nom d'utilisateur et d'un mot de passe). En effet, la reconnaissance du rôle opérationnel passe par cette phase d'identification.

Nous considérons l'outil comme l'agrégation de deux éléments : le dispositif matériel et l'application logicielle. Comme nous l'avons fait dans le méta-modèle précédent (MMPM), nous incluons un type « autre » dans chaque énumération dans l'éventualité ou celles-ci ne suffiraient pas à définir l'information voulue.

Nous distinguons plusieurs types d'application. Les applications dites natives sont dépendantes du système d'exploitation (OS) sur lequel elles sont exécutées alors que les applications web sont exécutées sur serveur et accessibles via un navigateur web depuis n'importe quel dispositif. Les applications hybrides utilisent les deux modes de fonctionnement. Dans tous les cas il sera nécessaire de déterminer la technologie de développement utilisée.

Nous définissons le dispositif matériel par plusieurs éléments. Nous pouvons inclure ici le type « inconnu » car il se peut que l'information ne soit pas accessible dans un cas de figure particulier. On identifie donc :

- un « type_dispositif » : ordinateur, tablette, smartphone, autre…
- un ou plusieurs « type_OS » (fonction du type de dispositif) : windows, macOS, linux, windowsphone, iOS, android, autre…
- les contrôleurs de type : clavier + souris, clavier + touchpad, écran tactile, autre…
- s'il y a une connexion à internet : oui/non/inconnu
- s'il y a une entrée vidéo : oui/non/inconnu
- s'il y a une entrée audio : oui/non/inconnu
- s'il y a une sortie vidéo : oui/non/inconnu
- s'il y a une sortie audio : oui/non/inconnu

Chaque usage (pour un collecticiel comme pour tout autre outil) est caractérisé par sa temporalité (c.à.d. exécuté à des dates précises, souvent ou pas, à intervalles réguliers…) et sa localisation (qui peut déterminer certaines conditions d'usage comme un environnement bruyant par exemple). Considérant les usages pendant un projet collaboratif de conception/construction architecturale, nous pouvons typer ces caractéristiques par des valeurs connues qui pourront être étoffées au fil de l'évolution de l'approche :

- le type de temporalité : plusieurs fois par jour, tous les jours, tous les 2-3 jours, une fois par semaine, une fois par mois, moins d'une fois par mois, après chaque réunion, autre/inconnu…
- le type de localisation : au bureau, sur le chantier, au domicile, chez un client, dans un endroit public, autre/inconnu…

Les relations entre usages

En comparant plusieurs usages, on pourra spécifier la relation entre les deux par ses dimensions spatio-temporelles :

- la synchronisation, qui détermine s'ils sont effectués en même temps (synchrones) ou pas (asynchrones),
- la colocalisation, qui détermine s'ils sont effectués au même endroit (colocalisés) ou à distance.

On dira que la temporalité et la localisation de chaque usage « déterminent » la relation entre plusieurs usages (Figure 100).

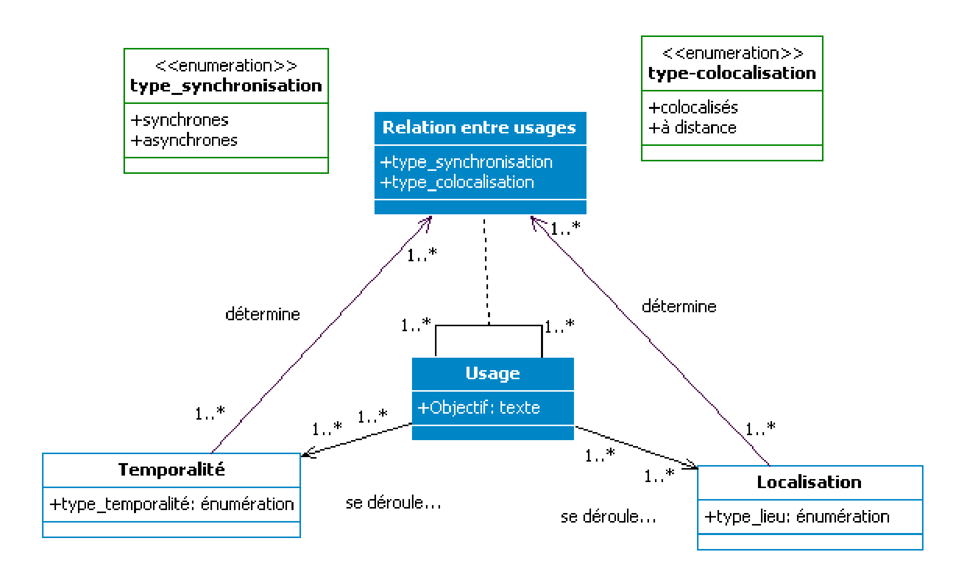



*Figure 100. Caractérisation de la relation entre usages*

### *1.32.4 Conclusion : le méta-modèle d'usage (MMU)*

L'association des méta-modèles que nous venons de définir compose le méta-modèle d'usage (MMU) (Figure 101). Le MMU est lié au méta-modèle de pratiques métiers (MMPM) par les différentes relations que nous avons également définies. Conceptuellement, nous caractérisons ainsi la médiatisation des pratiques par les usages. Comme le montre la section suivante, l'instanciation de ce méta-modèle se fait en plusieurs temps et à travers plusieurs formalismes.

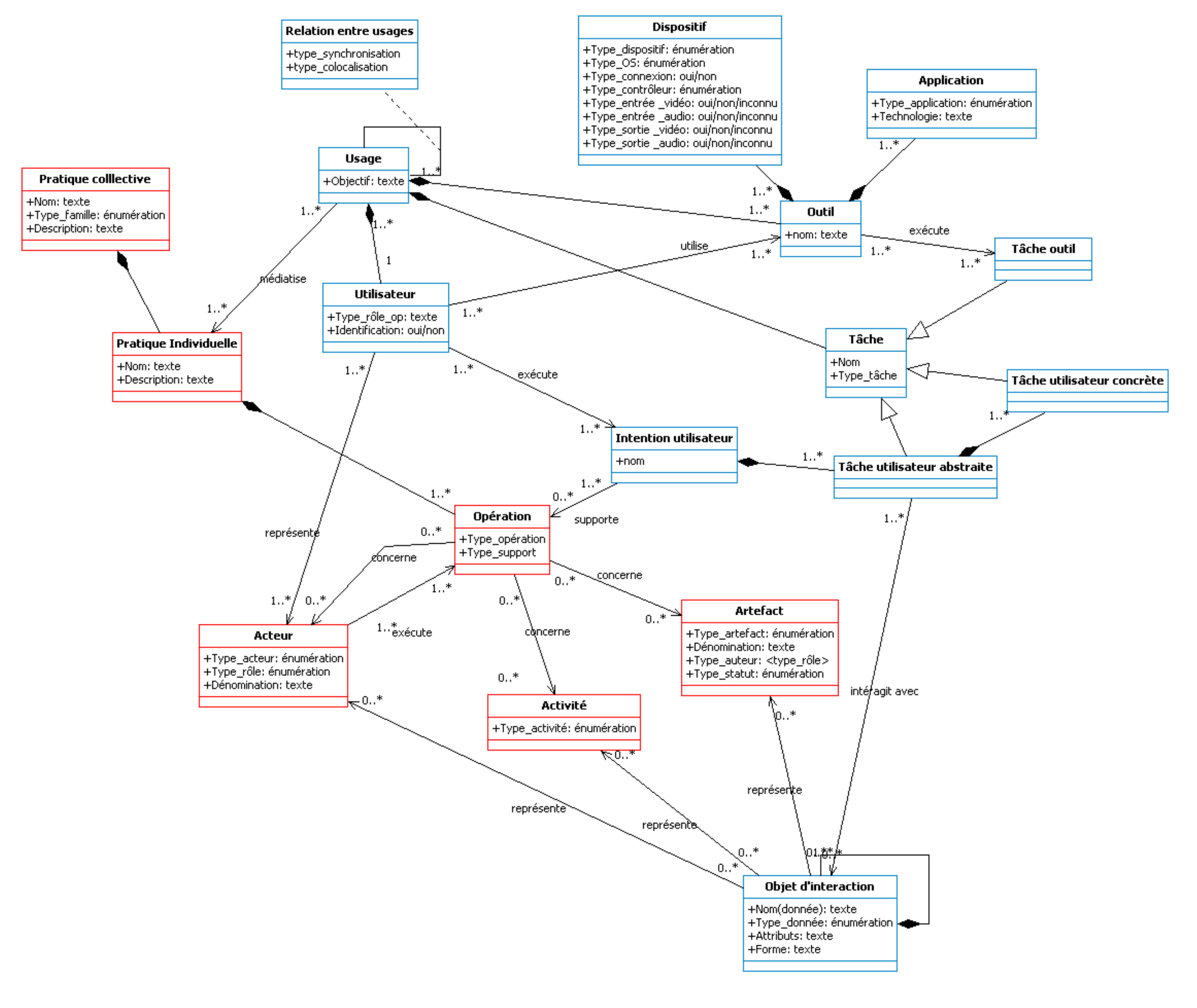



*Figure 101. Représentation simplifiée du MMU (bleu) lié au MMPM (rouge)*

## 1.33 Les modèles d'usage

Le MMU est un méta-modèle qui caractérise des concepts tirés du domaine du Génie Logiciel et de la conception d'IHM. Les formalismes utilisés pour l'instancier seront eux aussi relatifs à ces deux domaines. Ainsi, le point de vue opérationnel sera exprimé par les concepteurs à travers des concepts et des formalismes qu'ils comprennent et qu'ils ont l'habitude de manipuler. Ces formalismes sont les suivants :

- le diagramme de cas d'utilisation,
- le diagramme de tâches,
- le maquettage d'interfaces utilisateurs,
- et des tableaux.

### *1.33.1 Le « diagramme de cas d'utilisation contextualisé »*

Nous utilisons le diagramme de cas d'utilisation (Figure 102) pour modéliser dans un premier temps toutes les **intentions de l'utilisateur** et les réponses attendues de l'outil (**les tâches outil**). Elles sont modélisées par des use case différenciés graphiquement. Les paquets rouges sont

utilisés pour représenter l'opération dont sont déduites les intentions utilisateurs. Les paquets situés hors du cadre de l'outil représentent les opérations qui ont été identifiées dans la pratique mais qui ne sont pas l'objet du travail de conception en cours. Comme nous l'avons dit précédemment, tout cas d'utilisation n'est pas directement lié à une opération métier, comme par exemple les cas de configuration (s'identifier, paramétrer…). Ces cas d'utilisation qui doivent pourtant être spécifiés sont distingués par un package bleu.

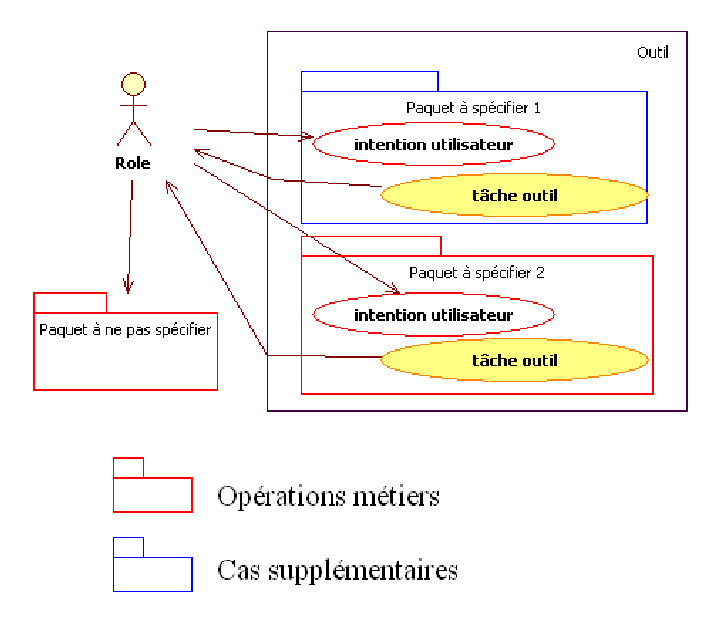



*Figure 102. Adaptation du formalisme des diagrammes de cas d'utilisation*

À ce niveau d'analyse, nous décrivons également le contexte de l'usage. Ici le formalisme utilisé est un tableau éditable par listes de choix multiples ou par champs d'édition libres. La composition de ces deux modèles (c'est à dire le diagramme de cas d'utilisation et le tableau), forme ce que l'on a appelé le « diagramme de cas d'utilisation contextualisés » (en anglais « contextual use case »).

Le « cas d'utilisation contextualisé » illustré ci-après (Figure 104) représente un exemple d'usage particulier d'un utilisateur de CRTI-weB pour accomplir une pratique que nous avons modélisée dans le chapitre précédent. La pratique considérée ici est « diffuse les documents de conception à l'équipe de conception » (Figure 103). Au cours de cet usage l'utilisateur souhaite partager plusieurs fichiers (intention « upload multiple files ») mais attend de l'outil qu'il groupe les fichiers choisis en un seul document CRTI-weB (tâche outil « upload files as one CRTI-weB document »). Ce nouvel usage varie de l'usage « traditionnel » qui consiste à créer un document CRTI-weB pour chaque fichier partagé. Il s'agit d'un cas simple mais qui reflète bien l'adaptation à un besoin en termes d'usage de la part d'utilisateurs dans leur contexte de projet.

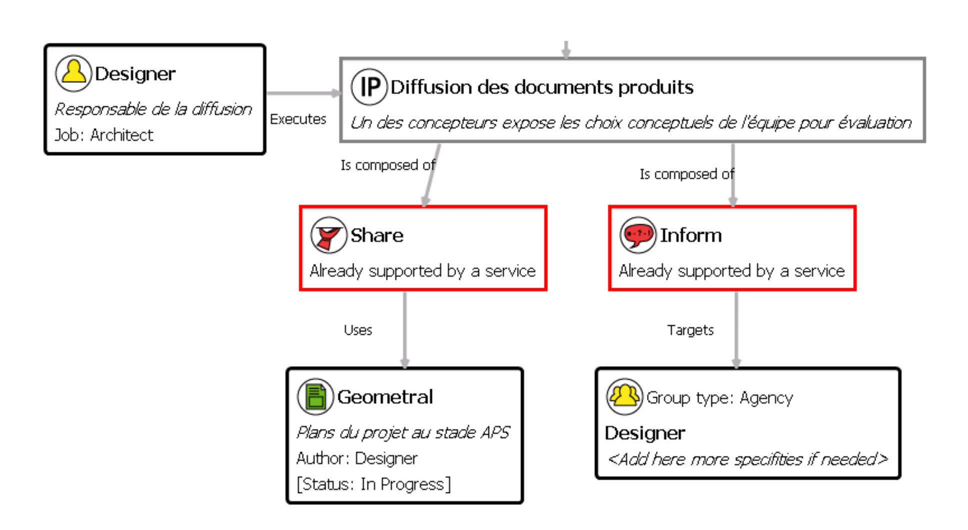
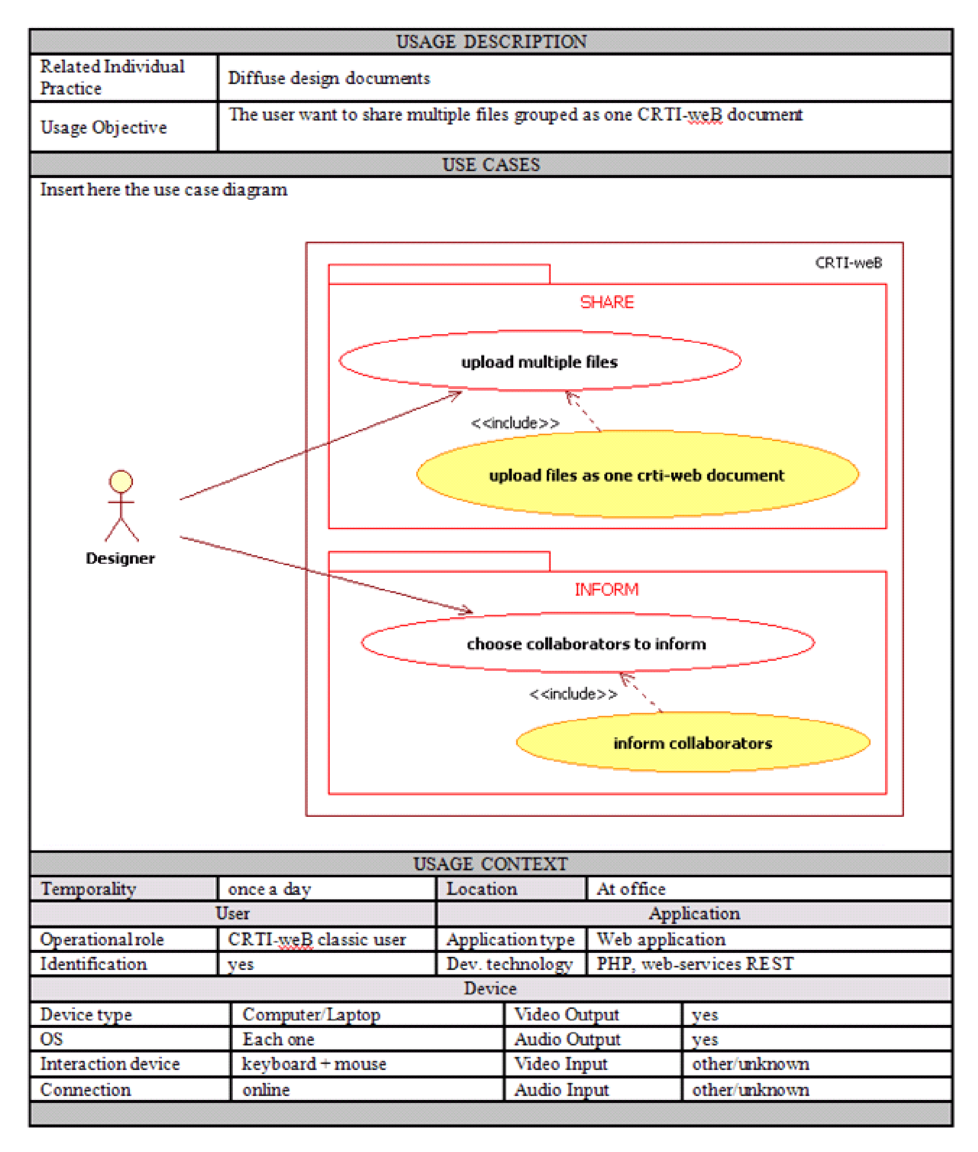



*Figure 103. Rappel : modélisation de la pratique « diffusion des documents »*



| USAGE DESCRIPTION | |
|---|---|
| Related Individual Practice | Diffuse design documents |
| Usage Objective | The user want to share multiple files grouped as one CRTI-weB document |
| USE CASES | |
| Insert here the use case diagram | |



| USAGE CONTEXT | | | |
|---|---|---|---|
| Temporality | once a day | Location | At office |
| User | | Application | |
| Operational role | CRTI-weB classic user | Application type | Web application |
| Identification | yes | Dev. technology | PHP, web-services REST |
| Device | | | |
| Device type | Computer/Laptop | Video Output | yes |
| OS | Each one | Audio Output | yes |
| Interaction device | keyboard + mouse | Video Input | other/unknown |
| Connection | online | Audio Input | other/unknown |

*Figure 104. « Contextual Use Case » de la pratique « diffuse design documents for designers team »*

La temporalité de l'usage est identifiée comme  effectué environ une fois par jour. On imagine en effet qu'à la fin d'une journée de travail, un concepteur veille à partager ce qu'il a produit. Le lieu de l'usage est l'agence dans laquelle il travaille.

Il y a plusieurs rôles opérationnels définis sur la plateforme CRTI-weB (service document) :

- l'utilisateur classique qui peut utiliser tous les services de l'outil (partage de documents, téléchargement, ajout d'actions, etc...),
- le superviseur projet : il peut en plus relancer des actions (par exemple les demandes de validation) auxquelles on n'a pas répondu,
- l'administrateur du projet : il peut ajouter des nouveaux utilisateurs au projet et créer des nouveaux projets.

Ici nous considérons l'utilisateur comme un utilisateur classique (classic user) dont le rôle organisationnel est celui de concepteur (designer). L'usage de la plateforme demande que l'utilisateur s'identifie. L'application et le dispositif sont également définis conformément à la caractérisation du méta-modèle.

Les relations entre usages

La modélisation des relations entre l'usage spécifié et d'autres usages prend la forme d'un tableau dans lequel on renseigne pour chacun d'eux : l'identifiant du cahier d'exigences qui les spécifie, l'objectif de l'usage, la synchronisation dans le temps (synchrone, asynchrone) et la co-localisation (co-localisé, à distance). Dans l'exemple ci-dessous (Figure 105) on considère l'usage qui consiste à récupérer les documents partagés. Par rapport à l'usage que nous venons de décrire, l'exécution peut se faire de manière asynchrone et à différents endroits.

| RELATED USAGES | | | |
|---|---|---|---|
| You can specify the link between the usage you described and others | | | |
| Form ID | Usage description | Synchronization | Co-location |
| X.X | Designers get documents from others | asynchrone | different locations |
| Click here to enter text. | Click here to enter text. | Choose an item. synchrone asynchrone | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |

*Figure 105. Modélisation de la relation avec d'autres usages*

### *1.33.2 Les diagrammes d'interaction et maquettages d'interfaces*

Pour modéliser les autres niveaux de tâches, nous nous sommes inspirés des modèles de tâches références (CTT et K-MAD). Après avoir utilisé K-MAD dans une première expérimentation, nous avons identifié la nécessité de proposer un formalisme personnalisé (Figure 107). Le choix de ce nouveau formalisme a été guidé par deux motivations :

- l'une était relative à l'aspect graphique : pour assurer la correspondance avec les modèles précédents, nous voulions pouvoir déterminer notre propre « charte graphique »,
- l'autre était relative au contenu : nous voulions modéliser, en plus des tâches, les objets d'interactions caractérisés dans notre méta-modèle d'usage.

Nous nous sommes à nouveau servis du framework GMF d'Eclipse pour générer l'éditeur de ce nouveau diagramme que nous appelons diagramme d'interactions. Un diagramme d'interactions décompose un « contextual use case ».

Dans ce formalisme, chacune des **intentions** de l'utilisateur identifiées dans le use case devient la racine (1$^{er}$ niveau) d'un arbre composé de **tâches utilisateurs abstraites** (2e niveau) puis **concrètes** (3e niveau).

Les **tâches outils** apparaissent également. Elles sont définies comme des tâches abstraites et sont également décomposées en tâches concrètes. Ces tâches concrètes sont relatives à la perception du « feedback » qui rend compte de l'exécution de la tâche outil (ex. l'affichage d'une notification ou l'émission d'un son).

Les propriétés graphiques des intentions utilisateurs et tâches outil sont identiques à celles utilisées dans le diagramme de cas d'utilisation : ellipse au contour rouge pour l'intention et ellipse au fond jaune pour la tâche outil. Cela permet d'assurer visuellement la correspondance entre les concepts, d'un modèle à un autre. Les tâches utilisateurs sont représentées par des ellipses au contour jaune. Les tâches concrètes sont représentées par des rectangles gris (Figure 106).

Nous complétons cette structure traditionnelle de l'arbre des tâches par les **objets d'interaction** définis précédemment. L'objet d'interaction caractérise la donnée manipulée ainsi que la forme que prend cette dernière. Ils apparaissent dans le diagramme sous forme d'un rectangle noir aux coins arrondis.

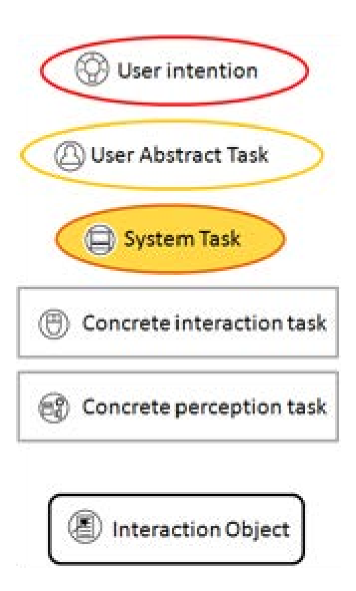



*Figure 106. Légende du diagramme d'interactions*

Le diagramme suivant (Figure 107) spécifie le use case modélisé précédemment qui formalise l'usage de CRTI-weB dans lequel l'utilisateur souhaite partager plusieurs fichiers et attend du l'outil qu'il groupe les fichiers choisis en un seul document. Cette intention « envoyer plusieurs fichiers » (« upload multiple files ») est décomposée en deux tâches abstraites : 1) le choix des

fichiers incluant le choix du regroupement puis 2) le nommage. Le choix des fichiers se définit par le suivi des tâches concrètes suivantes (rectangles gris sur la figure) :

- activation de la fonction « nouveau chargement »,
- sélection des fichiers,
- visualisation du chargement,
- sélection du regroupement,
- et enfin de validation de la sélection.

La tâche outil est spécifiée par une tâche concrète de perception « identification de la notification ».

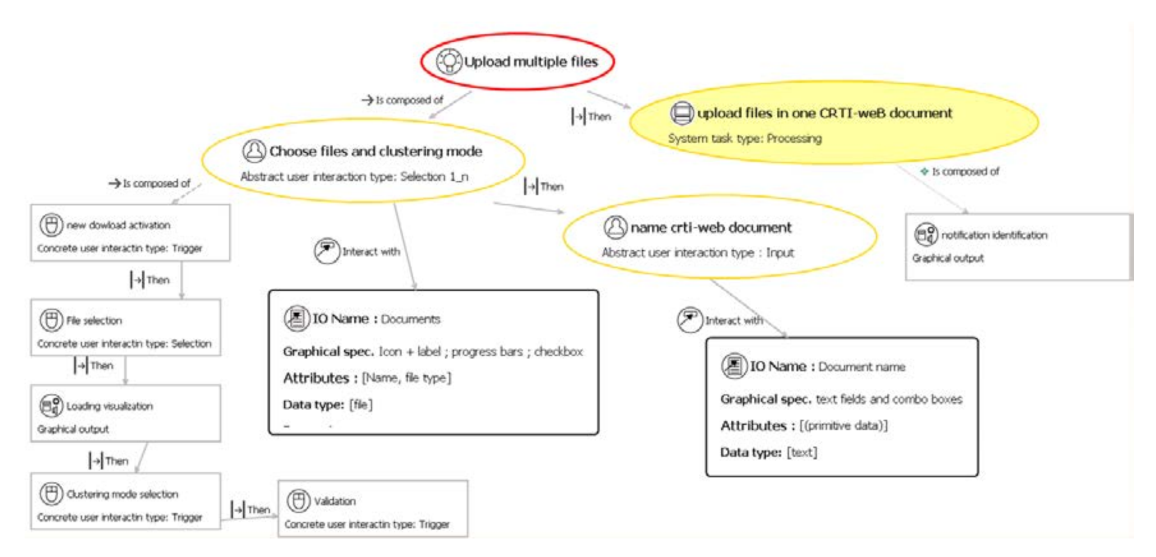



*Figure 107. Diagramme d'interaction décomposant l'intention « upload multiple files»*

<u>Les maquettages</u>

Le maquettage permet de représenter l'interface attendue à un instant précis sans perdre de temps dans les détails graphiques. Le maquettage est un mode de modélisation complémentaire au diagramme d'interactions pour instancier une partie de notre méta-modèle d'usage. Il permet en effet d'illustrer les tâches concrètes qui ont été spécifiées comme le montre la Figure 108.

Le risque lié maquettage est de le faire intervenir trop tôt dans le processus de conception. Cela a pour conséquence de limiter la conception d'un système à celle d'une interface graphique composée de fonctionnalités souhaitées mais dont l'adaptation aux besoins ne peut être validée. L'intégration du maquettage après une étude des pratiques métiers suivie d'une identification des intentions et tâches utilisateurs liées permet de limiter les propositions « à ce qui est nécessaire ».

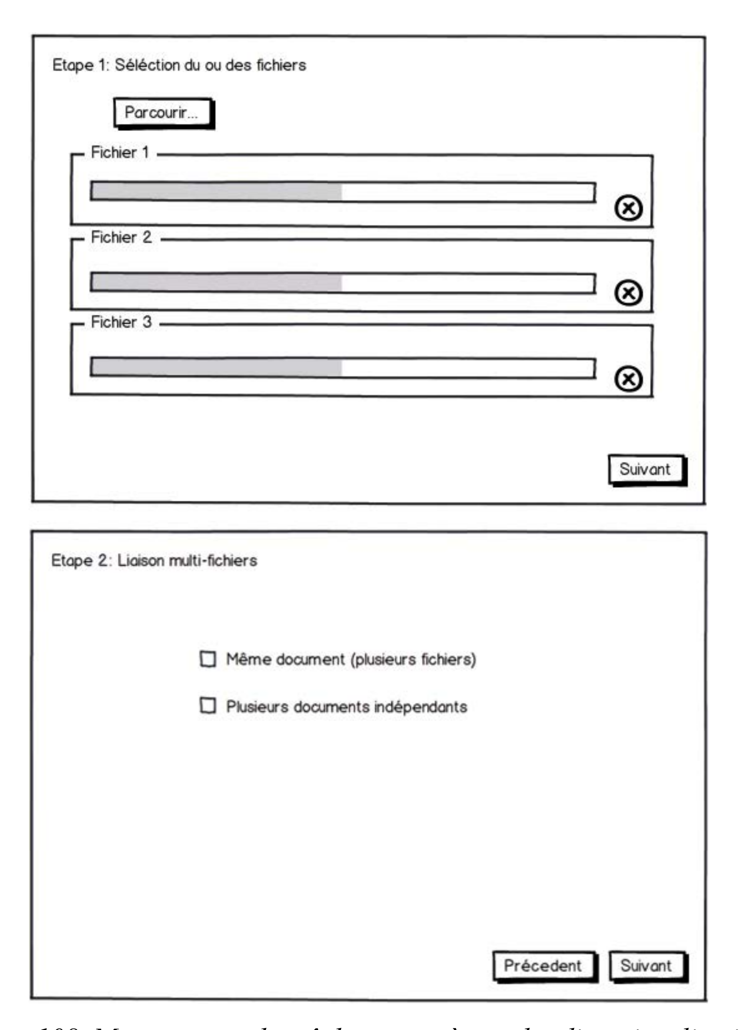



*Figure 108. Maquettages des tâches concrètes « loading visualization » et « clustering mode selection»*

### 1.33.3 Intégration dans le cahier d'exigences

La modélisation de l'usage est formalisée dans la seconde partie du cahier d'exigences. Le « diagramme de cas d'utilisation contextualisé » et le tableau spécifiant les relations avec d'autres usages sont insérés dans un premier temps. Comme lors de l'analyse des pratiques métiers, il est également possible d'ajouter de l'information additionnelle. Le but est de laisser aux concepteurs la possibilité de consigner les informations qu'ils jugeront utiles mais que les modèles formalisés n'auraient pas permis de prendre en compte. Ces informations supplémentaires pourront être raffinées au fur et à mesure des versions du cahier d'exigences.

Ces informations additionnelles prennent la forme d'un tableau dont les parties visent à compléter la description du contexte de l'usage, de l'utilisateur, et de l'outil (application + dispositif).

On inclut de la même manière dans la deuxième partie du cahier d'exigences le diagramme d'interaction et les maquettages produits.

| ADDITIONAL INFORMATION ABOUT CONTEXT |
|---|
| Do you want to add any functional or non-functional requirements about… |
| The general context of the usage? |
| You can describe here the pre-conditions and post-conditions of the usage. It can refer to the relations with other usages considering the time (synchronous, asynchronous) and the location (same location, different location).<br>Click here to enter text. |
| The actor's specificity as a user? |
| What do you know about the actor using the system? Eventually precise here, his skills, knowledge, abilities, preferences…<br>Click here to enter text. |
| The application specificity? |
| Is there already a developed application to adapt? You can describe it here.<br>Click here to enter text. |
| What about the new services to develop?<br>Click here to enter text. |
| The device specificity? |
| Does it have a particularity in terms of mobility, performance, autonomy, capacity…? Do you think about a specific model?<br>Click here to enter text. |
| |

*Figure 109. Tableau de saisie des informations supplémentaires dans le cahier d'exigences*

## 1.34 Conclusion

La caractérisation des pratiques permettait d'appréhender « ce que les acteurs d'un projet font », celle de l'usage est dédiée à la compréhension de « comment ils le font ». Il s'agit ici de décrire :

- l'utilisateur et l'outil,
- le contenu médiatisé manipulé,
- les tâches effectuées autour de ce contenu, par les utilisateurs et l'outil.



*Figure 110. Le choix d'un usage et sa définition*

La conceptualisation, formalisée par des méta-modèles sous forme de diagrammes de classes UML, nous a permis de lier conceptuellement les aspects organisationnels (relatifs aux pratiques) aux aspects opérationnels (relatifs aux usages). Ce lien se retrouve dans l'instanciation de ces méta-modèles à travers l'adaptation des formalismes utilisés à cet effet (Figure 111).

Ainsi, le diagramme de cas d'utilisations est directement conçu pour répondre aux opérations métiers identifiées. Il est complété par diverses informations de contexte, ce qui en fait un modèle pertinent pour décrire la médiatisation d'une pratique par l'usage.

Le travail sur les tâches d'interaction peut paraître superflu sur un cas simple comme celui présenté. Le diagramme de cas d'utilisation aurait suffi à décrire l'usage attendu. Il pourra cependant être plus utile dans d'autres cas d'étude, comme nous le verrons dans le chapitre 12 de ce manuscrit. Dans un Framework de MDA (Model-driven Architecture) comme Caméléon introduit au § 1.11.2, le premier niveau du processus (le CIM) définit les tâches utilisateurs et les concepts manipulés. Même si cela n'est pas exploité ici, la définition du diagramme d'interactions pourra donc s'inscrire dans ce genre d'approches génératives. Cela pourra faire l'objet de travaux futurs.

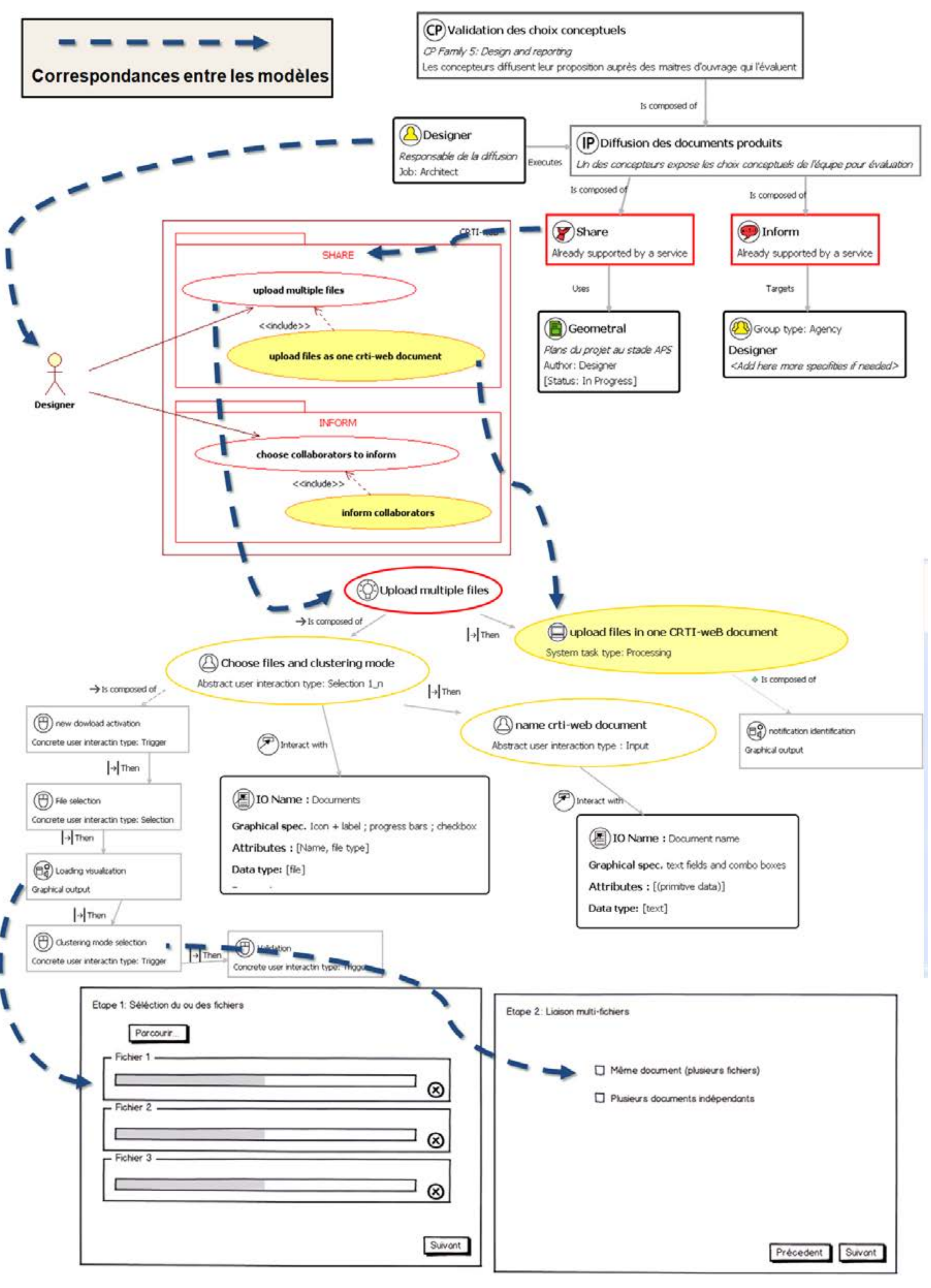



*Figure 111. Correspondances entre le diagramme de pratiques et les modèles d'usage (diagramme de cas d'utilisation, d'interactions et maquettages)*

Alors que le travail sur les pratiques est un travail d'analyse, s'amorce ici le travail de conception. En effet, nous caractérisons le contexte attendu, les intentions ainsi que les scénarii d'interactions idéaux. En d'autres termes, il s'agit ici de concevoir l'usage idéal, sans préoccupations d'ordre technique et donc sans pour l'instant appréhender les cas d'erreurs et les limites technologiques possibles. Les sociétés de services jouent ce rôle de concepteur en tirant parti du travail des experts métiers qui interviennent dans l'analyse des pratiques. Le passage d'un point de vue à un autre sur la base de modèles doit assister cette collaboration.

La section suivante illustre comment la description de l'usage nous guide vers la spécification de services, d'un point de vue purement fonctionnel. Dans la continuité de notre approche, nous conceptualisons le service en fonction des concepts de l'usage pour assurer la correspondance entre les modèles proposés. Cette étape n'est autre que la fin du processus de conception de services : alors qu'à travers l'usage on perçoit l'outil comme « une boite noire » (en décrivant abstraitement les tâches de l'outil en réponse aux tâches de l'utilisateur), la description des services proposés détaille la solution telle qu'elle devra être implémentée. Tel que nous l'avons introduit, plusieurs usages alternatifs peuvent correspondre à une même pratique. La prise en compte de cette variation dans les usages nous permet de répercuter cette diversité dans les services proposés pour convenir ainsi à une majorité des utilisateurs finaux.

Atteindre ce niveau de détail est nécessaire pour les développeurs. En effet, cela permet de percevoir toutes les limites qu'il faudra surmonter pour développer la solution, ce qui pourra faire varier la conception de celui-ci. Le dialogue et la compréhension entre concepteurs et développeurs sont la clé de cette troisième étape.

# Chapitre 11 – La modélisation des services : le point de vue fonctionnel

Le point de vue fonctionnel porte sur la description des services qui matérialisent les usages définis précédemment. Ce chapitre définit le concept de service tel qu'utilisé dans notre méthode et le caractérise par un méta-modèle de service (MMS) qui est mis en correspondance avec le méta-modèle d'usage (MMU). Cette correspondance définit la matérialisation des usages par les services tout comme la correspondance entre le MMU et le MMPM (Méta-modèle des pratiques métiers) caractérise la médiatisation des pratiques par les usages. Dans un projet de conception et développement de service, le point de vue fonctionnel est celui porté par les développeurs qui implémentent la solution pour atteindre l'usage attendu. Cette solution est modélisée, par instanciation du MMS, et de façon à supporter le dialogue avec les concepteurs notamment lorsqu'il s'agit de rendre compte des limites et contraintes techniques rencontrées. Les échanges entre concepteurs et développeurs peuvent avoir un impact sur le point de vue opérationnel, générant ainsi des itérations dans le déroulement de la méthode.

## 1.35 Définitions et concepts

### *1.35.1 Services métier et services informatiques ICT*

La première partie du chapitre 5 de cette thèse a été dédiée à l'étude du concept de service. Sur la base des définitions introduites alors, nous exprimerons le sens qu'adopte le concept de service dans notre méthode. Nous pourrons ensuite le caractériser.

Pour rappel, nous avons défini de manière générale un service comme une prestation qui met à disposition d'un client « une capacité technique ou intellectuelle » pour supporter un besoin. Nous distinguons deux types de services : les services métier et les services informatiques ICT (Services informatiques de Technologies de l'Information et de la Communication).

Nous avions déjà évoqué cette distinction au chapitre 2 lors de la description de l'outil CRTI-weB. En effet, celui-ci propose deux services dits métiers que sont le service « documents » et le

service « comptes-rendus », décomposés en services informatiques ICT comme la rédaction, la génération d'un pdf, la consultation… (cf. 1.4.3). Le terme service métier est ici emprunté au domaine de l'entreprise et réinterprété : il n'est plus échangé entre deux parties opérantes d'une entreprise mais il est offert par un outil pour répondre à un besoin métier. L'utilisation du terme service informatique respecte quant à lui le sens qu'on a pu lui associer, notamment dans (Kohlborn et al. 2009) : c'est une fonctionnalité logicielle qui supporte l'exécution d'un service métier.

Nous garderons ces deux définitions à l'esprit pour caractériser le point de vue fonctionnel, troisième et dernière étape de la méthode PUSH pour la proposition de services adaptés.

### *1.35.2 Caractérisation des services*

En respectant le vocabulaire utilisé dans le domaine de l'entreprise orientée services, nous caractérisons un service par : un nom, une description, un client et un fournisseur.

La description d'un service comprend normalement à la fois les propriétés fonctionnelles (capacités du service, paramètres pouvant varier) et les propriétés non fonctionnelles (qualité dans la fourniture du service, prix…). Ces dernières ne faisant pas l'objet de ce travail, notre description concernera essentiellement les capacités et paramètres du service.

Le fournisseur propose au client le service dont il a besoin. Dans notre cas, le fournisseur n'est autre que l'outil considéré, le client représentant alors l'utilisateur de cet outil.

(Kohlborn et al. 2009) évoque deux caractérisations des relations qui ont lieu au cours de l'exécution du service :

- le modèle interactionnel qui caractérise l'interaction avec le client par des entrées et des sorties de données (les Inputs et les Outputs),
- le modèle opérationnel qui décompose le service en un processus d'actions (ou sous-services)

Le méta-modèle proposé dans la section suivante formalise ces quelques éléments de caractérisation du service. Nous y incluons :

- la caractérisation de la « matérialisation » d'un usage par un service métier
- la composition d'un service métier en services informatiques ICT
- la décomposition d'un service ICT suivant un modèle architecture logicielle

## 1.36 Méta-modèle de service

### *1.36.1 Le service : matérialisation de l'usage*

Rappelons qu'à cette étape de la conception de notre méthode :

- nous connaissons l'organisation professionnelle d'un projet AIC caractérisée par le méta-modèle du contexte coopératif (MMCC),
- nous avons caractérisé les comportements métiers sous forme de pratiques (collectives et individuelles) et d'opérations à partir des concepts de ce MMCC et en créant le méta-modèle des pratiques métiers MMPM),
- nous avons caractérisé la médiatisation de ces pratiques par les usages d'outil dédiés, (définissant l'outil, l'utilisateur et l'interaction qui a lieu entre les deux dans leur contexte particulier) en créant le méta-modèle d'usage (MMU) à partir des concepts du MMPM .

Il s'agit maintenant de caractériser la matérialisation des usages par les services en créant le méta-modèle de services (MMS, Figure 112) dérivé du MMU. Conformément à la description que nous en avons faite plus haut :

- le service métier matérialise l'usage, il est caractérisé par un nom et une description,
- son fournisseur représente l'outil,
- il est composé de services informatiques ICT,
- le client de ces services représente l'utilisateur de l'outil,
- la relation client-service n'est autre que l'interaction entre l'utilisateur et outil, précédemment définie par les tâches et objets d'interaction.

Les inputs et outputs sont des flux de données en entrée et sortie. Ils sont définis par une action effectuée et une donnée manipulée. Les inputs correspondent aux tâches concrètes d'interaction et les outputs aux tâches de perception. La donnée a été définie au travers du concept d'objet d'interaction.

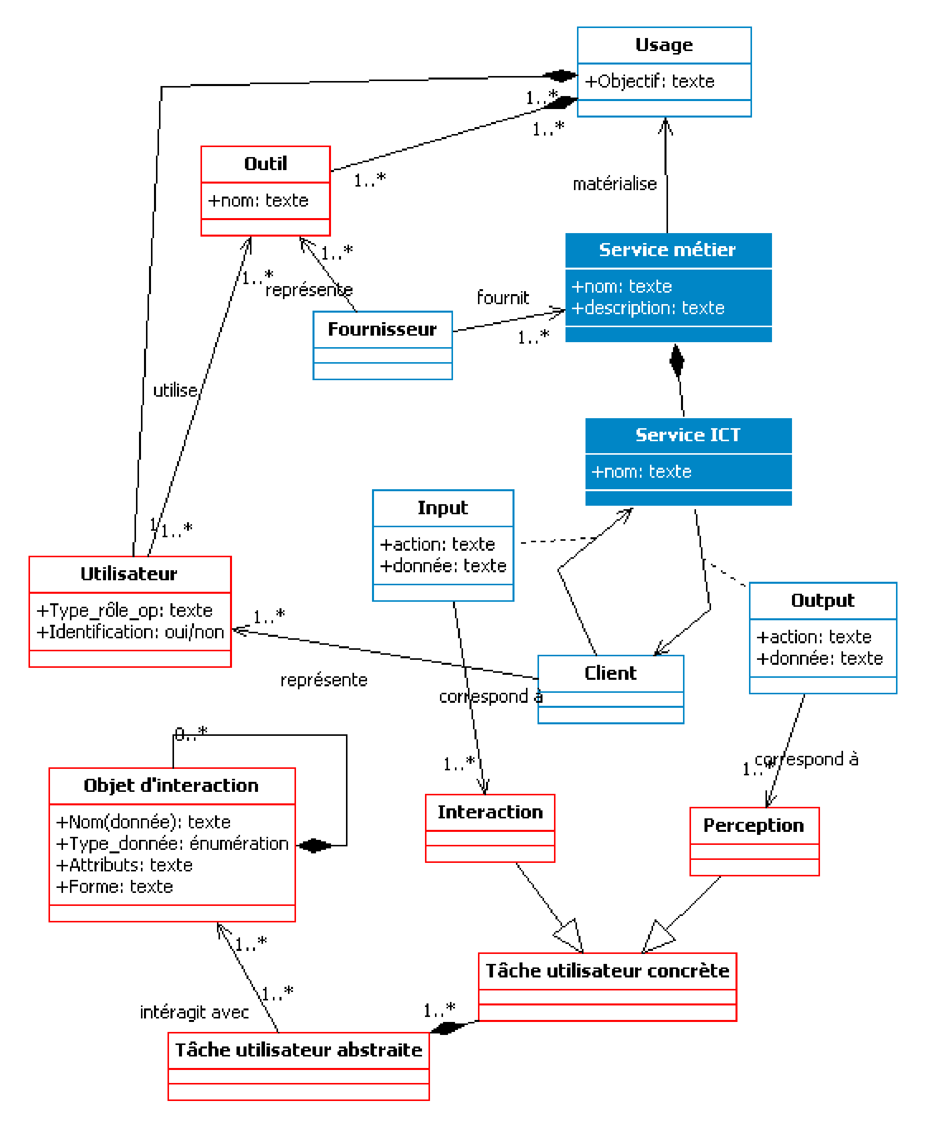



*Figure 112. Caractérisation du service (concepts en bleu) matérialisant l'usage (concepts en rouge)*

### *1.36.2 Les composants du service ICT*

Nous décrivons le service ICT comme un ensemble de composants logiciels qui communiquent entre eux. Cette architecture peut être caractérisée par un modèle d'architecture logicielle.

Nous avons introduit précédemment les modèles d'architecture logicielle et particulièrement ceux dédiés aux collecticiels (cf § 1.16.3). Nous avons étendu l'un deux, le modèle MVC, pour y inclure la composante collaborative, créant ainsi le modèle Co-MVC (Figure 113). Ainsi, dans la caractérisation du service :

- le Modèle caractérise les données et les méthodes pour manipuler ces données (par exemple ajouter, supprimer, récupérer…). Le modèle privé concerne la gestion des données pour chaque utilisateur (ex. sur son ordinateur personnel). Il se synchronise avec le modèle partagé qui met en commun les données pour tous les utilisateurs,
- la Vue est l'interface avec laquelle l'utilisateur interagit. Par le biais de fenêtres, d'icônes, etc… elle capte les interactions de l'utilisateur (par exemple les clicks) et lui renvoie des informations,
- le Contrôleur caractérise les composants logiciels qui transmettent les requêtes depuis la vue, transmettent les données, interrogent les bases de données, réactualisent la vue… Ce sont des applications, des services-web… On peut parler ici de sous-services.

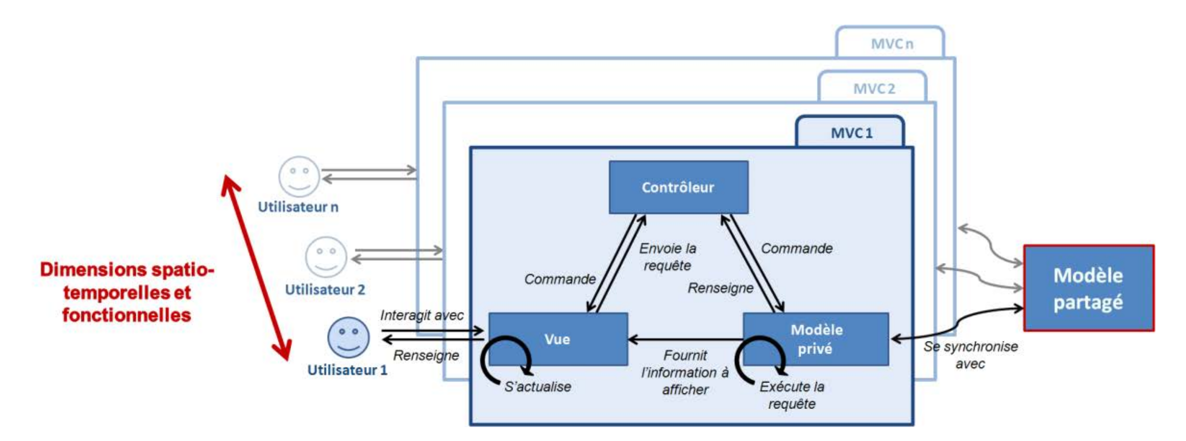



*Figure 113. Rappel : le modèle Co-MVC*

Ainsi, le méta-modèle de services (Figure 114) caractérise le service ICT comme un ensemble de ces trois types de composants : la Vue, le Contrôleur, le Modèle spécifié par le modèle privé et le modèle partagé. Chacun d'eux est caractérisé par un nom et la technologie de développement utilisée (ex. une interface JAVA (vue), un web-service REST (contrôleur)…). Les échanges qui s'opèrent entre eux sont des actions manipulant des données, au même titre que les Inputs et Outputs.

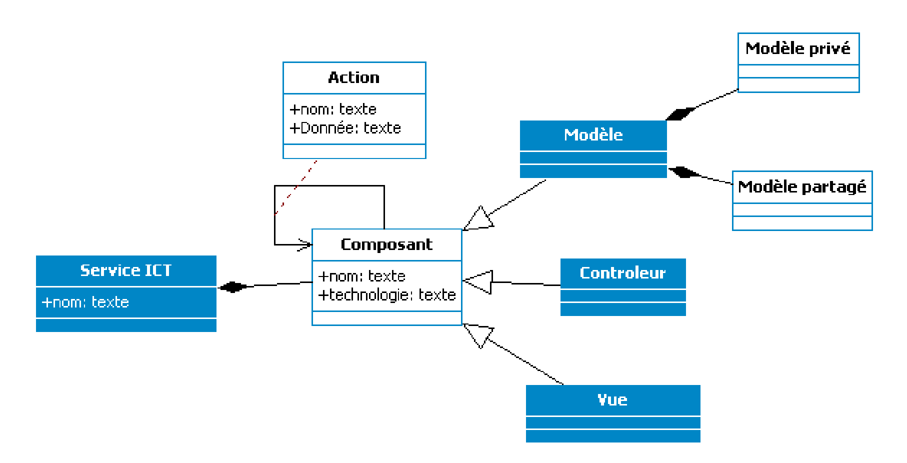



*Figure 114. Caractérisation du service ICT et ses composants*

### *1.36.3 Le méta-modèle des services ICT adaptés*

Par la composition des deux méta-modèles présentés dans cette section, nous créons notre Méta-Modèle de Services ICT (MMS) de la même façon que nous avons composé les méta-modèles de pratiques et d'usages (MMPM et MMU). Ce que nous appelons Méta-Modèle des Services Adaptés (MMSA) est l'association de ces trois méta-modèles. En effet, partant du postulat qu'un service est adapté s'il répond à des besoins métiers, cette association qui caractérise la médiatisation des pratiques par les usages, eux-mêmes matérialisés par les services caractérise donc cette adaptation.

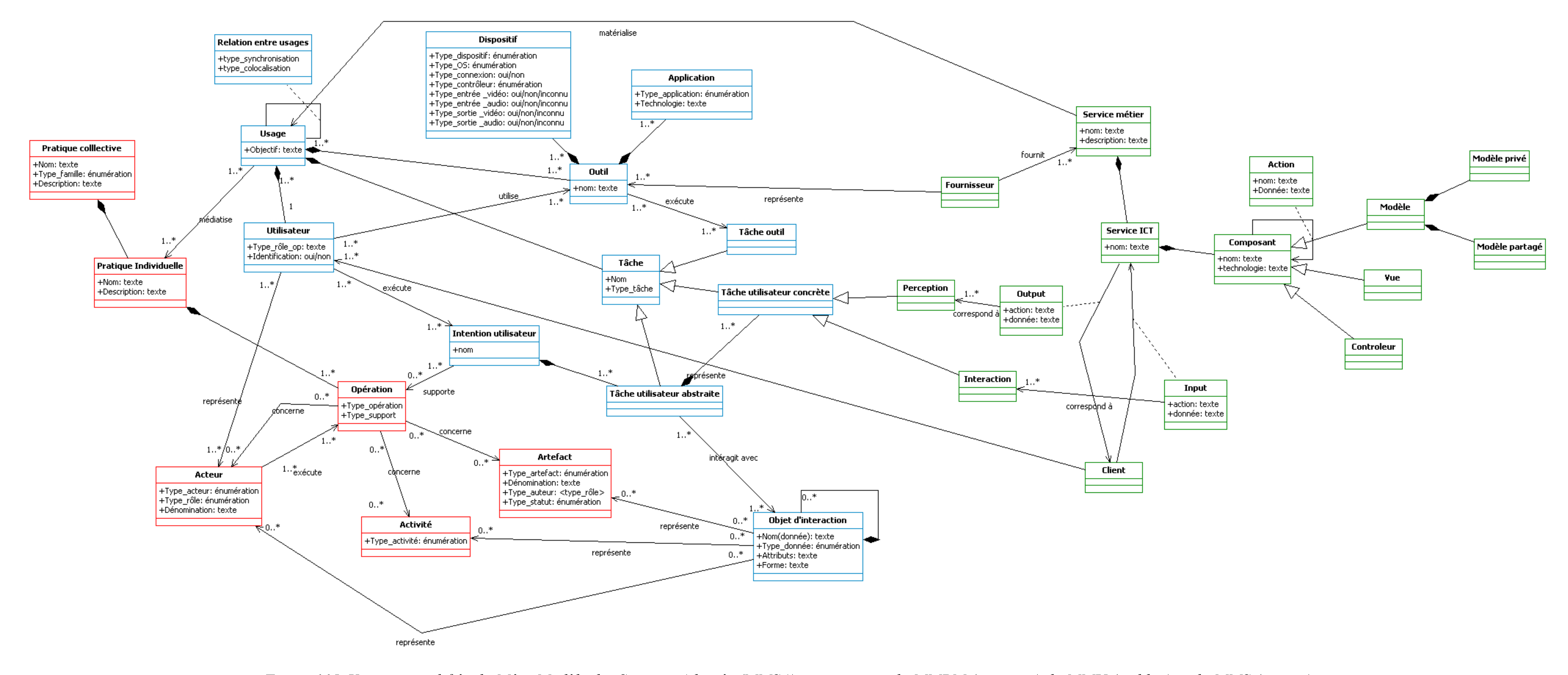


*Figure 115. Version simplifiée du Méta-Modèle des Services Adaptés (MMSA), composition du MMPM (en rouge) du MMU (en bleu) et du MMS (en vert)*

## 1.37 Modélisation des services et implémentation

Nous utilisons le diagramme de séquences pour décrire le comportement du client et du service. Cette description doit être conforme au Méta-Modèle de Services qui intègre le modèle d'architecture Co-MVC. Nous adoptons pour cela le diagramme de séquences UML, un formalisme utilisé en conception logicielle pour décrire la solution dans une architecture logicielle et matérielle (Len Bass et al. 2003) (cf. § 1.9.3, Figure 24).

### *1.37.1 Le diagramme de séquence*

Il s'agit d'instancier ici le Méta-Modèle des Services. Nous représentons sur la base d'un diagramme de séquences UML : l'outil qui fournit le **service métier**, la description de ce service métier, le **client**, le **service ICT** considéré, ses **composants** et enfin les **échanges** (inputs, outputs et actions) qui s'opèrent.

Reprenons l'exemple du partage de documents pour illustrer cette étape. Il s'agit d'effectuer ce partage via l'outil CRTI-weB, service (métier) documents (voir § 1.33). Le diagramme de tâches qui a été créé lors de la définition de l'usage est repris ci-dessous (Figure 116).

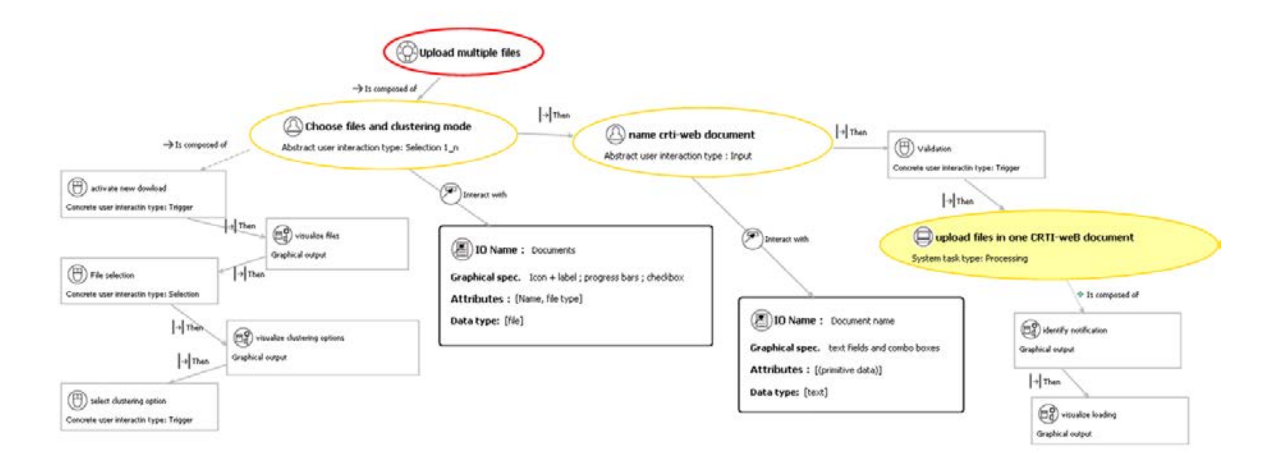



*Figure 116. Rappel : Diagramme d'interaction décomposant l'intention « upload multiple files»*

Les tâches concrètes d'interaction et de perception (rectangles gris) guident la spécification des échanges client-service. Il s'agit alors de définir le comportement du service ICT en décrivant les actions de ses composants (Figure 117).

Nous sommes dans un cas d'adaptation d'un service déjà existant. La solution technique adoptée est donc dépendante de la technologie utilisée et s'inscrit dans les développements déjà effectués. Les développeurs, responsables de ces développements, sont à même d'exprimer ce point de vue fonctionnel pour décrire ce qu'ils vont implémenter.

L'utilisateur manipule des fichiers « physiques » (stockés sur son ordinateur et qu'il souhaite copier sur un serveur externe). Lorsqu'il a choisi les fichiers qu'il veut partager, le service informatique les copie sur le serveur sous forme d'un fichier ZIP. Puis il crée dans la base de données un document CRTI-weB auquel il associe le fichier ZIP qu'il vient de créer.

Techniquement, cette association se fait par la définition de l'attribut « localisation » du document CRTI-weB avec l'URL du fichier ZIP créé sur le serveur.

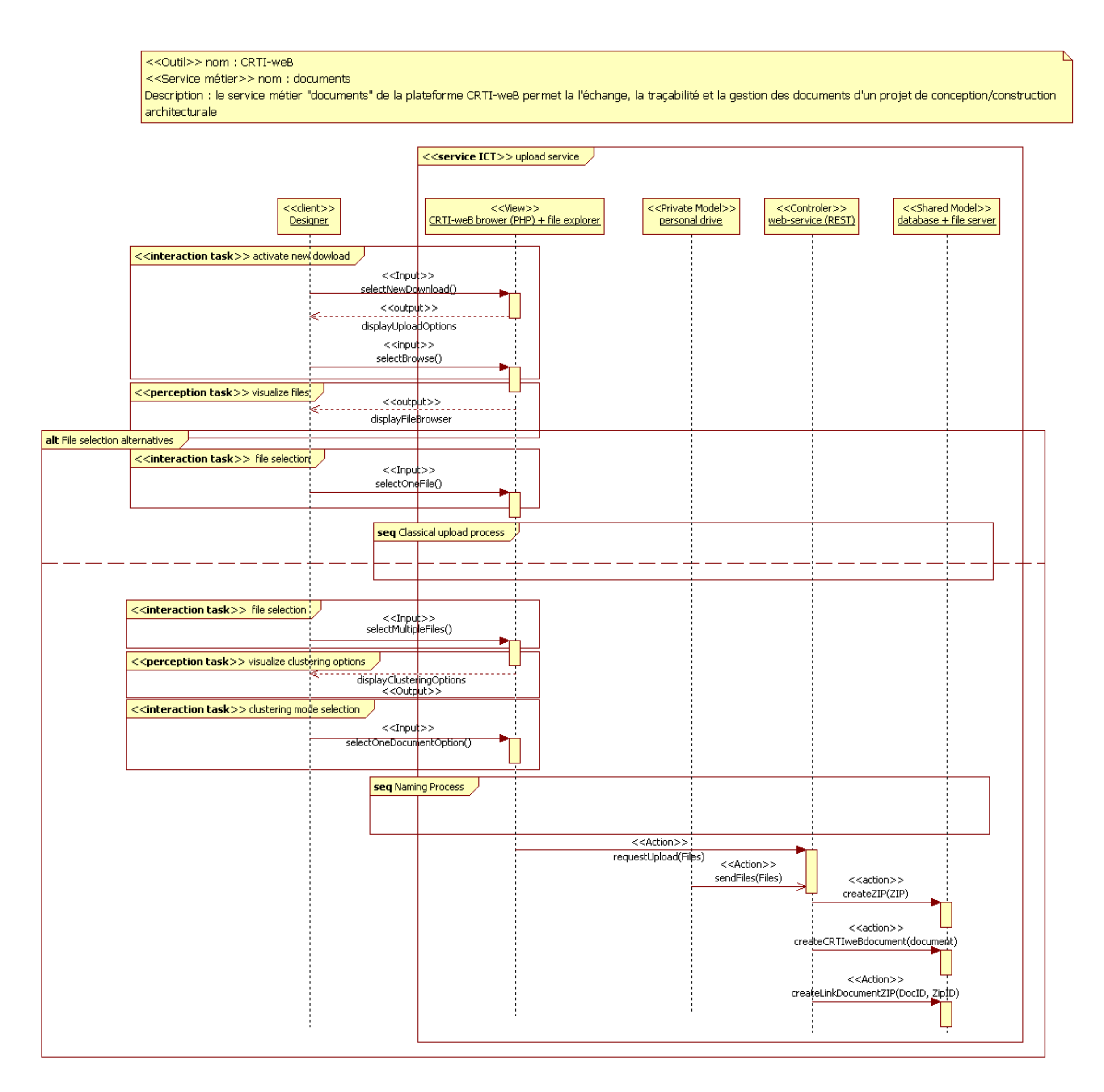



*Figure 117. Diagramme de séquence modélisant le service ICT de téléchargement multi-fichiers via l'outil CRTI-weB (service métier document)*

### *1.37.2 Intégration dans le cahier d'exigences*

La dernière partie du cahier d'exigences est éditée lors de l'expression du point de vue fonctionnel par la modélisation des services. Cette phase est directement suivie par les premières activités de développement et se réitère pendant celles-ci. En effet, même si le service a été spécifié correctement, il est courant que les développeurs et les concepteurs doivent revoir certains détails (comme la gestion des cas d'erreurs par exemple).

L'ajout de remarques et commentaires sur les diagrammes de séquences permettra de rendre compte des spécificités du service. Les outils d'édition de modèles UML (comme StarUML)

permettent cette expression de commentaires sous forme de « notes » qui peuvent être associées à des éléments du diagramme.

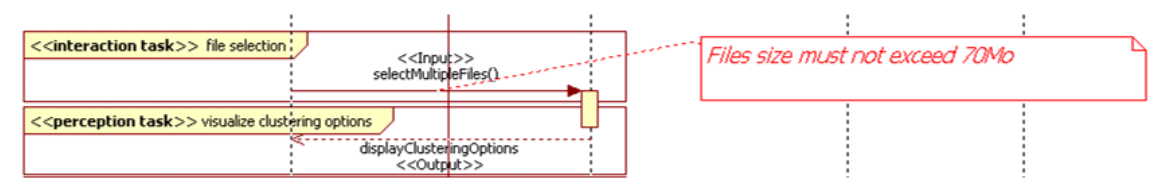



*Figure 118. Exemple de note ajoutée à un Input dans un diagramme de séquences modélisant un service*

Dans certains cas (si par exemple le concepteur n'avait pas réellement conscience des limites techniques de l'outil), il se peut que le scénario d'usage ne puisse pas être matérialisé tel qu'il a été conçu. La modélisation d'un service alternatif permettra de supporter le dialogue entre les développeurs et les concepteurs qui devront à terme repenser leur usage.

Lors des modifications, qu'elles soient relatives aux modèles de services ou aux modèles d'usage, la création d'une nouvelle version mise à jour du cahier d'exigences permettra :

- de formaliser les décisions prises afin de pouvoir s'y référer,
- et de garder une trace de l'évolution du projet et ainsi justifier, à posteriori, les changements entrepris.

### *1.37.3 Développements*

Le processus lié à l'implémentation des services spécifiés ne fait pas l'objet de cette étude, l'objectif de la méthode proposée étant de formaliser des exigences qui serviront de base aux développements.

Le suivi et le compte-rendu des développements par les concepteurs et développeurs peuvent se faire sur base du cahier d'exigences (cf. section précédente). Au fil des développements, des prototypages et démonstrations peuvent être présentés pour tester le service en cours d'implémentation. Des tests du service final sont effectués par les concepteurs voire par les futurs utilisateurs, avant le transfert du service. La gestion des tests n'est pas décrite dans la méthode PUSH. Certaines méthodes (agiles) préconisent la rédaction des tests en même temps que les besoins (cf. § 1.9.4).

La figure suivante présente quelques captures d'écran du service décrit précédemment une fois implémenté. Y sont représentés : le choix d'un nouveau partage (cliquer sur « ajouter un nouveau document »), le parcours des fichiers (cliquer sur « parcourir » puis sélectionner les fichiers), le choix du lien (cocher l'option « les documents sont liés »). Le résultat est, comme spécifié, le partage d'un fichier zip contenant les documents.

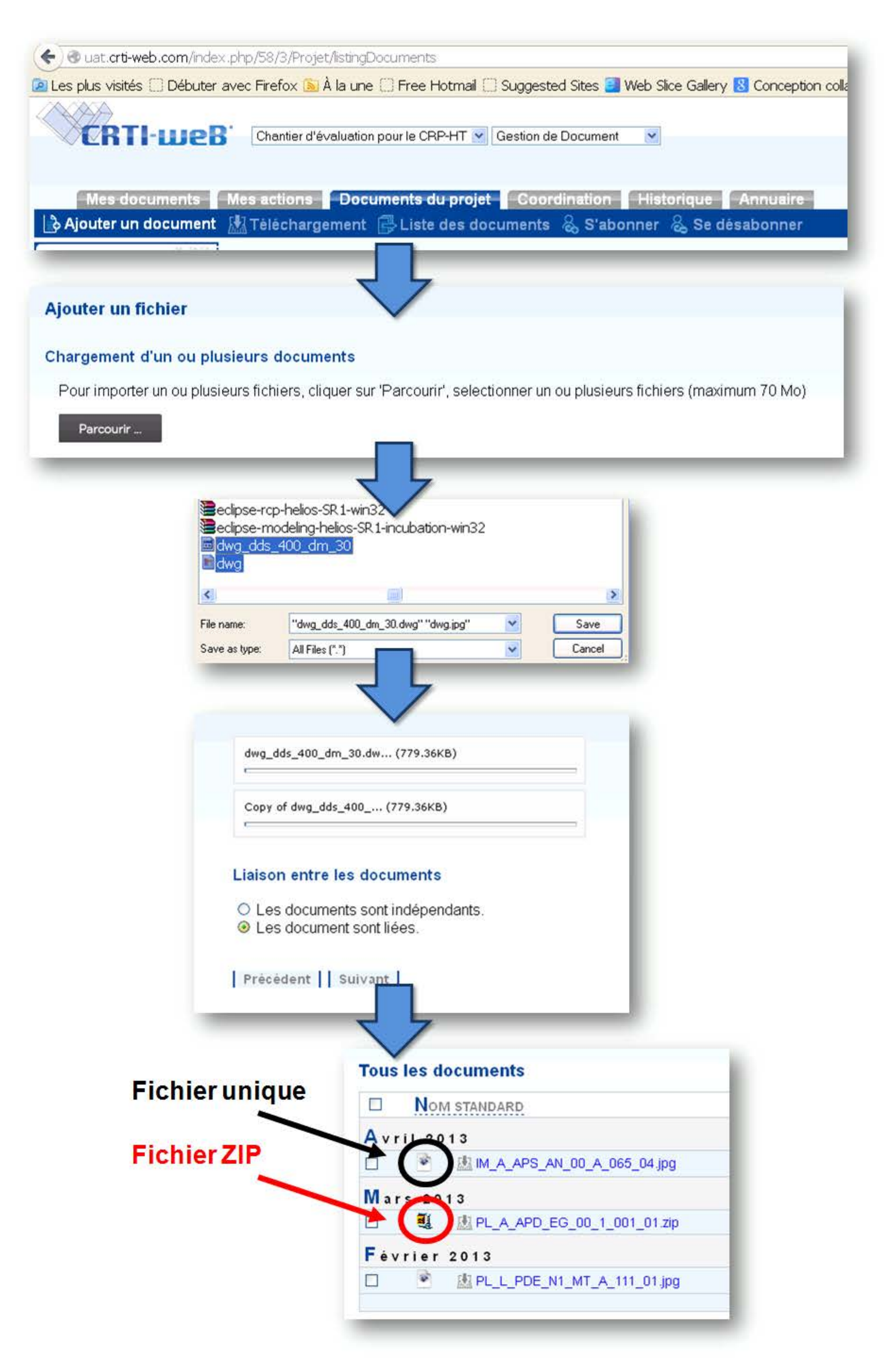



*Figure 119. Illustration de l'utilisation du service ICT de téléchargement multi-fichier une fois implémenté (cf. diagramme de séquences Figure 117)*

## 1.38 Conclusion

Le méta-modèle des services caractérise la partie fonctionnelle qui servira de base au développement. À partir des besoins identifiés en termes de pratiques et d'usages, cette troisième phase de modélisation spécifie le service qui sera le plus adapté. Les liens entre le méta-modèle de services et le méta-modèle d'usages assurent l'adaptation des services développés par rapport aux attentes des concepteurs.

Le diagramme de séquence est un formalisme utilisé par les développeurs et constitue le support du dialogue entre ces derniers et les concepteurs. L'évolution des spécifications, reportée dans les cahiers d'exigences, trace les choix de développement (point de vue fonctionnel) et éventuellement les retours sur la conception de l'usage (point de vue opérationnel).

L'approche de conception de services orientée usage permet des spécifications précises qui ciblent des besoins clairement identifiés. C'est une approche qui s'applique particulièrement bien dans le cadre de l'amélioration d'un outil existant. La base du développement est en effet déjà construite, ce qui permet de limiter les spécifications techniques en réutilisant l'existant. Cependant, cette base existante peut également devenir une limite car elle rend certains choix impossibles à implémenter. L'itération du processus est nécessaire pour adapter ces choix aux limites. C'est pourquoi il est important de privilégier la collaboration et le dialogue entre concepteurs et développeurs pour contourner les problèmes qui pourraient se présenter lors d'une phase plus avancée du projet.

À terme, nous pensons que la combinaison d'un ensemble d'exigences ainsi spécifiées permettra de concevoir un système entier. L'uniformisation des choix par des concepts récurrents et structurés est la clé de cette combinaison.

L'exemple utilisé pour illustrer les trois étapes de la méthode (chapitres 9-10-11) est un cas simple qui nous a permis de présenter l'approche et les concepts développés. Cet exemple ne reflète cependant pas un besoin métier très particulier, l'usage associé et le service développé restant donc relativement communs. Nous illustrons plus précisément l'intérêt de cette méthode au cours du chapitre suivant, à travers trois expérimentations. À défaut de valider le processus entier à chaque fois, chacune de ces expérimentations nous permettra d'évaluer un aspect particulier de la méthode. Nous essaierons d'en tirer parti pour mener une analyse critique générale de l'approche.

# Chapitre 12 – La méthode PUSH : expérimentations et bilan

## 1.39 Introduction aux expérimentations

### *1.39.1 Rappel : application d'une méthode dirigée par les modèles*

Au travers les précédents chapitres (9-10-11) nous avons cherché à caractériser plusieurs points de vue différents portés sur le travail collaboratif. Adopter chacun de ces points de vue permet de construire, pas à pas, un méta-modèle qui décrit des services adaptés aux besoins métiers :

- le point de vue organisationnel porte sur les pratiques métiers
- le point de vue opérationnel porte sur les usages qui médiatisent ces pratiques
- le point de vue fonctionnel porte sur les services qui matérialisent ces usages




*Figure 120. Représentation conceptuelle du lien entre les points de vue*

Chacun des concepts (pratique, usage et service) est décrit par son propre méta-modèle (le MMPM, le MMU et le MMS). Les liens conceptuels « médiatise » et « matérialise » (Figure 120) sont définis par plusieurs correspondances entre ces trois méta-modèles et forment le Méta-Modèle des Services Adaptés (MMSA) (cf. Figure 115, page 197).

Ces correspondances sont identifiables graphiquement lors de l'instanciation du MMSA par l'adaptation des formalismes utilisés (Figure 121). Ces formalismes sont le diagramme hiérarchique pour le modèle de pratiques, le diagramme de cas d'utilisation puis l'arbre de tâches pour modéliser l'usage et enfin les diagrammes de séquences pour le service. Les transformations sont les suivantes :

- L’acteur de la pratique (diagramme de pratiques) devient utilisateur du système (diagramme de cas d’utilisation)
- Une opération métier (diagramme de pratiques) devient un ensemble d’intentions utilisateurs (package de cas d’utilisation dans le diagramme)
- Une intention utilisateur (diagramme de cas d’utilisation) devient la racine d’un arbre de tâches utilisateurs (diagramme d’interactions)
- Les tâches outils (identiques dans le diagramme de cas d’utilisation et le diagramme d’interactions) sont décomposées en actions des composants du service (diagramme de séquences)
- Les informations métiers (diagramme de pratique) deviennent des objets d’interaction (diagramme d’interaction)
- Les objets d’interaction (diagramme d’interaction) servent à spécifier les données échangées lors de l’exécution du service (diagramme de classes)

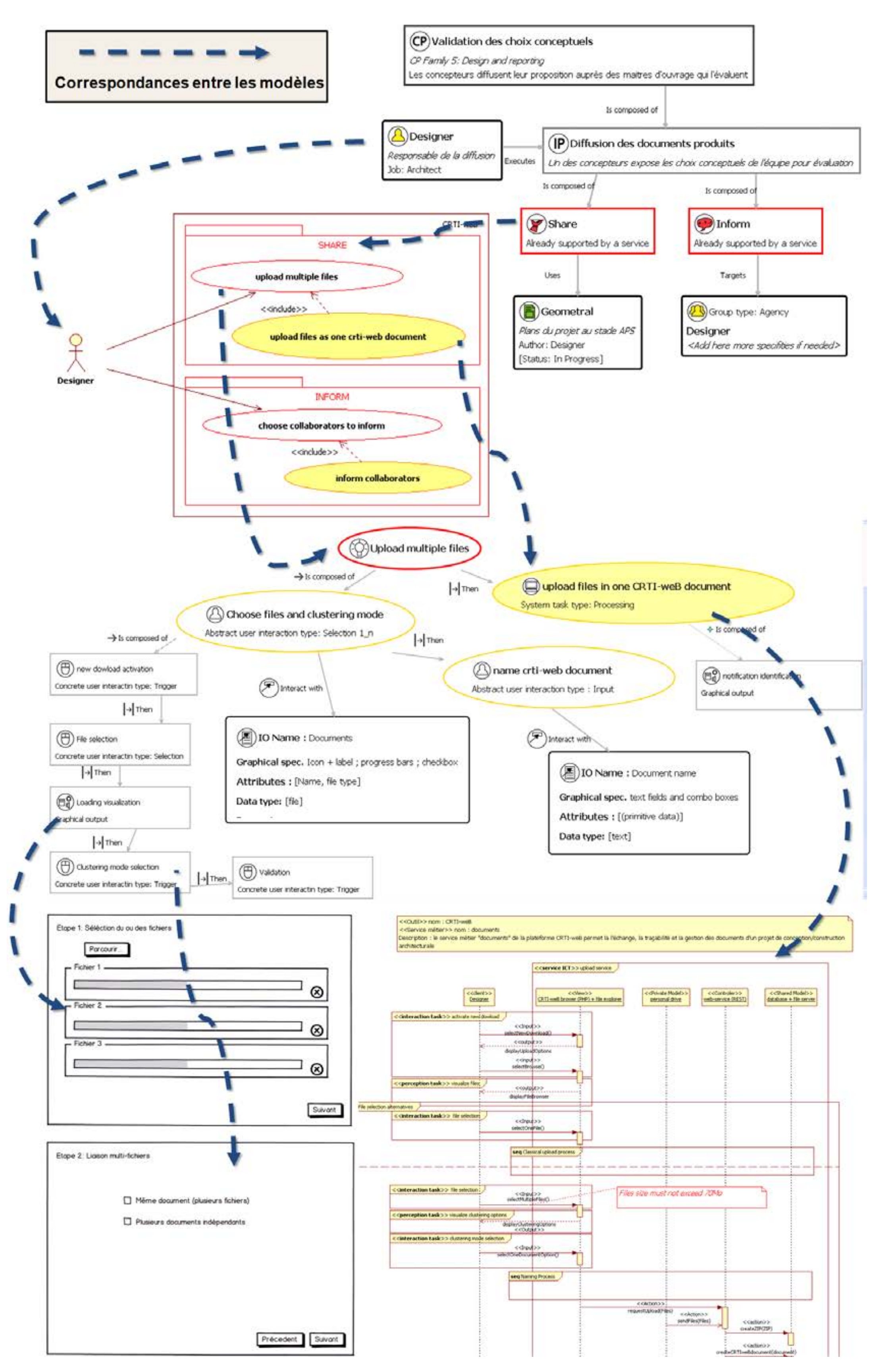



*Figure 121. Correspondances entre la modélisation des pratiques, des usages et des services*

### 1.39.2 *Contexte des expérimentations*

Pour rappel, notre travail s'est inscrit dans le cadre du développement de la plateforme de gestion documentaire CRTI-web. Les expérimentations menées se sont inscrites dans ce contexte technologique particulier : les deux premières expérimentations concernent l'amélioration du service métier « Documents » tandis que la troisième consiste à proposer un nouveau service métier, mais utilisant l'information traitée par le service métier « Comptes-rendus ».

La nature de la première expérimentation était l'application de la méthode PUSH pour le transfert d'exigences vers une société externe (Kitry consulting) qui est en charge de deux activités :

- les développements informatiques sur la plateforme
- le transfert vers les professionnels et l'accompagnement au travers d'ateliers.

Ce contexte partenarial nous a permis d'identifier les besoins en termes d'échanges entre l'utilisateur, le concepteur et le développeur dans un projet de développement logiciel dédié aux professionnels du secteur de la construction (Figure 122). Il nous a également donné matière à réflexion sur l'analyse des besoins en termes de pratiques métiers et d'usages des utilisateurs dans la recherche de solutions à des problèmes rencontrés par des professionnels.

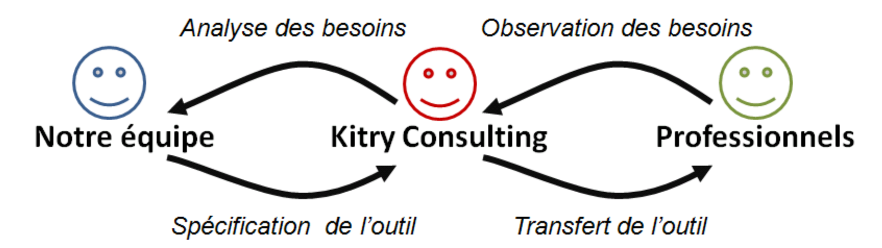



*Figure 122. Le contexte de projet de développement lors de la première expérimentation*

Deux autres expérimentations ont permis d'évaluer les hypothèses faites au cours de la première. La deuxième expérimentation a été menée en collaboration avec un autre informaticien jouant le rôle de développeur dans le projet, et visait à automatiser le service d'envoi de fichiers sur la plateforme CRTI-weB. La dernière était une expérience de conception seule, sans objectif de développement à court terme, visant à spécifier un service de surveillance de chantier supporté par un tableau de bord numérique sur tablette tactile.

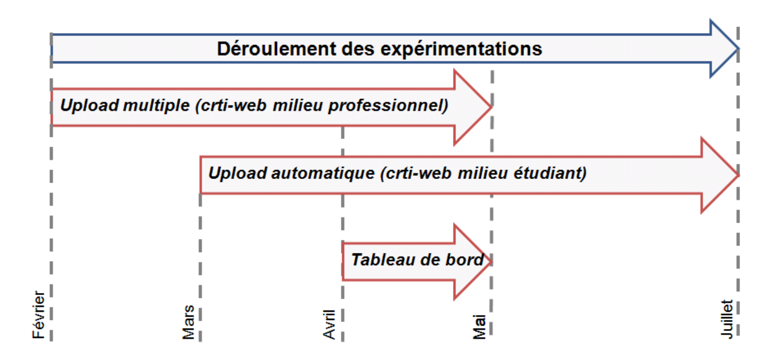



*Figure 123. Répartition des expérimentations dans le temps*

### *1.39.3 Objectifs des expérimentations*

Comme nous venons de l'expliquer ci-dessus, nous avons mené trois expérimentations de nature et de durées différentes. Elles ont permis de construire, évaluer et améliorer notre approche. Les objectifs à atteindre à travers les expérimentations étaient :

- justifier la méthode, en identifiant si au cours des expérimentations un réel besoin était ressenti et si la méthode y répondait,
- évaluer (et améliorer) la compréhension et l'utilisation des concepts introduits pour définir les étapes de la méthode (pratiques, usages, services),
- évaluer (et améliorer) la compréhension et l'utilisation des modèles à travers les outils et formalismes proposés,
- valider la réelle adaptation des services développés grâce à l'application de la méthode.

Comme le montrent les sections suivantes, chaque expérimentation ne portait pas sur l'ensemble de ces objectifs mais a fait ressortir des points d'intérêt particuliers sur lesquels s'appuyer pour évaluer et améliorer l'approche. Le tableau ci-dessous résume les attentes en termes d'évaluation que nous avions vis-à-vis de chaque expérimentation (en fonction de leur nature).

Tableau 24. Évaluations attendues en fonction des expérimentations

| Attentes… | Upload multiple | Upload Automatique | Tableau de bord |
|---|---|---|---|
| **cf. méthode** | Évaluer le besoin des développeurs | Évaluer l'importance des points de vue | Évaluer l'intérêt de structurer les exigences |
| **cf. concepts** | Évaluer l'analyse métier et la description de l'usage | Évaluer la distinction pratique/usages/ services | Évaluer la description des pratiques et de l'usage |
| **cf. modèles** | Évaluer les formalismes des pratiques et des usages | Évaluer les différents formalismes | Évaluer les formalismes de pratiques et d'usage ainsi que le passage d'un à l'autre |
| **cf. adaptation** | Évaluer l'adoption par les utilisateurs lors du transfert du service | Évaluer l'adoption des utilisateurs par des tests | Non concerné |

## 1.40 Expérimentation n°1 : L'amélioration du service d'upload de CRTI-weB

### *1.40.1 Introduction et déroulement*

Nous revenons ici plus en détail sur l'exemple « fil rouge » de conception d'un service « d'upload multiple » présenté au cours des chapitres précédents pour illustrer nos propos. Initié très tôt dans les phases de proposition de ce travail de recherche, il nous a servi à :

- composer les méta-modèles,
- choisir les formalismes pour les modèles,
- et définir la structure de la méthode PUSH.

Le besoin remonté au travers des ateliers avec des utilisateurs était celui de pouvoir partager plusieurs documents à la fois sur la plateforme CRTI-weB sans répéter le processus de dépôt d'un document depuis le début. Il s'agissait donc d'un besoin d'optimisation de la manière dont les

utilisateurs travaillent avec leur outil. De nombreux services de partage de documents permettent actuellement ce genre de manipulation. Cependant, il s'est révélé pertinent de mettre en place notre méthode afin d'identifier des exigences plus spécifiques au secteur que celles relatives à des services informatiques plus « généralistes ». Dans un contexte « marché », avec une approche commerciale, la démarche proposée devait permettre de lever le risque d'engager des coûts de développement dans une solution qui pourrait s'avérer inadaptée. Cet argumentaire a convaincu notre partenaire privé (la société de services externe) qui s'est prêté au jeu de l'expérimentation à travers notamment trois séances de travail (Figure 124).

Objet des séances de travail

La première s'est déroulée en février 2012. À partir des besoins exprimés lors des ateliers, nous avons présenté notre étude des pratiques relatives à ces besoins et de l'usage à concevoir pour y répondre (point de vue organisationnel et opérationnel). Les modèles présentés étaient un diagramme de pratiques, un diagramme de cas d'utilisations et un diagramme de tâches exprimé en K-MAD. Ces modèles ont permis respectivement de justifier le besoin métier (en montrant qu'il relève de pratiques courantes) et de définir les scénarios d'usage « idéaux ». Trois scénarios ont été identifiés. Ils étaient consignés dans une première version du cahier d'exigences. Au terme de cette réunion, les développeurs nous ont demandé de préciser le service à développer (point de vue fonctionnel). Un diagramme de séquences serait plus pertinent que le diagramme de tâches qui n'apporte pas assez de précisions fonctionnelles.

La seconde séance (mars 2012) a fait l'objet de discussions autour des diagrammes de séquence proposés. Il était question de repenser le service de partage de documents en fonction des trois scénarios d'usage imaginés et en spécifiant la gestion des cas d'erreurs. L'objet de la réunion était de valider les diagrammes proposés en y apportant éventuellement des modifications.

Les développeurs ont ensuite pu appréhender les possibilités d'implémentation du service spécifié. Au cours de la troisième séance, ils nous ont proposé des maquettages qui illustraient l'interface du service tel qu'il allait être développé. Au regard de certaines contraintes techniques, il y aurait quelques différences entre le service développé et sa spécification. Après validation des choix par tout le monde, le projet est entré en phase de développements. Le service de partage ainsi modifié a été implémenté sur une plateforme de tests dans un premier temps puis sur la version commercialisée.

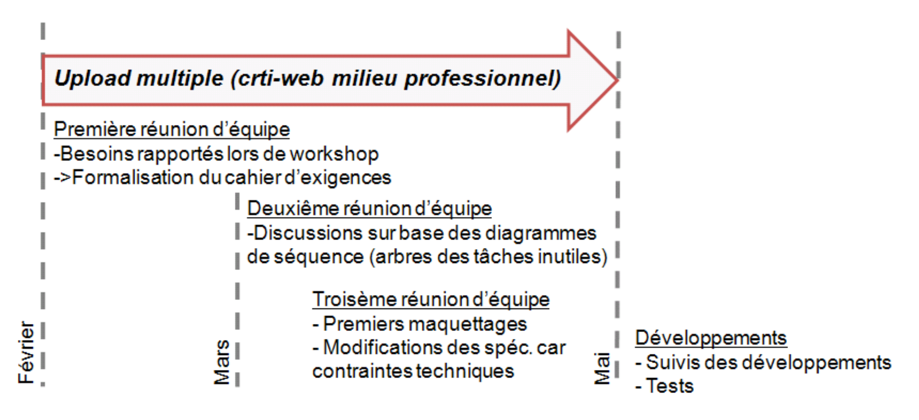



*Figure 124. Déroulement de l'expérimentation*

### *1.40.2 Les pratiques de partage*

Nous avons pu distinguer deux pratiques individuelles différentes réalisées par le rôle de « concepteur » (généralement l'architecte), chacune issue d'une pratique collective particulière pendant la phase de conception du projet :

- une première consistant à partager des plans et autres spécifications techniques avec les dessinateurs de l'agence pour créer les différentes vues du bâtiment (dans la Pratique Collective « échanges de document pour le partage de la conception »)une deuxième consistant à partager les plans et documents techniques avec l'entreprise de construction (dans la PC « échange des documents pour exécution »). Une pratique individuelle similaire serait celle du partage de documents avec l'économiste chargé de déterminer les coûts du bâtiment.

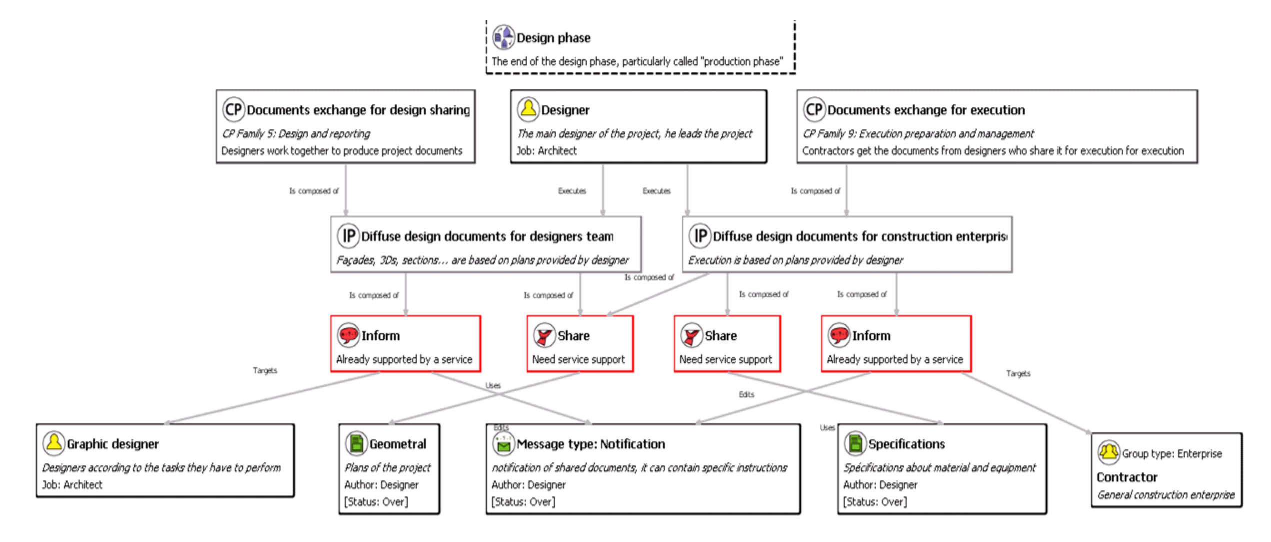



*Figure 125. Les pratiques de partage d'un concepteur*

Le modèle réalisé à partir de cette analyse métier (Figure 125), représente cette situation collaborative :

- la phase de projet est « la phase de conception »,
- les 2 PC considérées « échange de documents pour le partage de la conception «  et « pour l'exécution »,
- l'acteur « concepteur » auquel on s'intéresse,
- ses PI « diffuse les documents de conception pour l'équipe de concepteurs » et « pour l'entreprise de construction »,
- les opérations « partage » et « informe » pour chaque PI
- et enfin les artefacts manipulés à savoir les documents de type géométral (ex. les plans) et  de type spécifications (ex. cahier des charges), les messages de notifications ainsi que les autres acteurs impliqués (le dessinateur et l'entreprise de construction).

Nous observons comment, dans un même projet, la diffusion d'informations n'est pas un processus unique mais peut se décliner en plusieurs pratiques. Seules les observations faites en collaboration avec les professionnels du secteur permettent de capter ces spécificités. Il s'agit ici de les représenter au travers de cas réels et non pas de modéliser un processus général. Le formalisme utilisé supporte cette représentation.

Ce diagramme permet donc de comprendre comment se déroule un projet avec un certain degré de détail. Mais l'intérêt de cette distinction des pratiques réside dans la manière dont on va les médiatiser. En effet nous verrons que les usages peuvent varier en fonction de la pratique.

Hors ces variations qui assurent l'adaptation d'un outil et son appropriation par les utilisateurs, ne sont souvent pas prises en compte dans les démarches de conception « classiques ».

### *1.40.3 Les variantes d'usage*

Les professionnels utilisant CRTI-weB avaient relevé la nécessité de partager plusieurs documents d'un coup en simplifiant le processus d'envoi (déjà contraignant de par les multiples saisies relatives au nommage des documents et au choix de la zone et des actions). La première solution était d'implémenter un service générique permettant de choisir plusieurs fichiers à partager et de créer autant de documents CRTI-weB que de fichiers partagés. Mais au regard des usages spécifiques au métier, nous avons pu proposer une solution alternative adaptée à différents scénarios de pratique validés.

Deux pratiques, deux usages

À partir de l'analyse des pratiques, nous avons pu mettre en évidence que les professionnels peuvent vouloir diffuser des documents indépendants (PI diffusion aux autres concepteurs) ou préférer partager un groupe de documents avec un seul objectif (PI diffusion des plans d'exécution avec l'entreprise). À ces deux pratiques correspondent deux « intentions » en termes de médiatisation par un outil et donc deux usages.

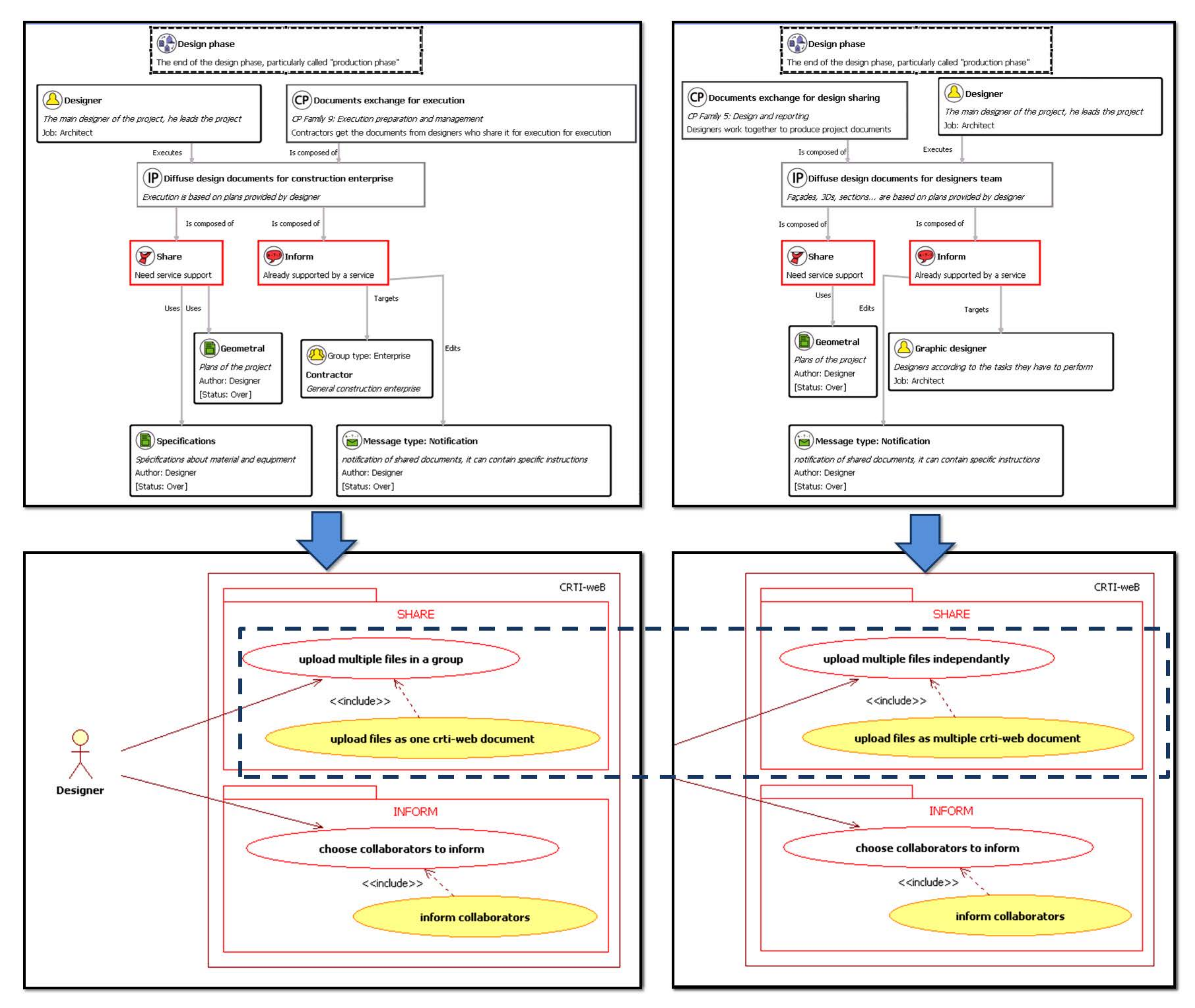



*Figure 126. De la pratique à l'usage : variation des intentions d'un utilisateur en fonction du besoin métier*

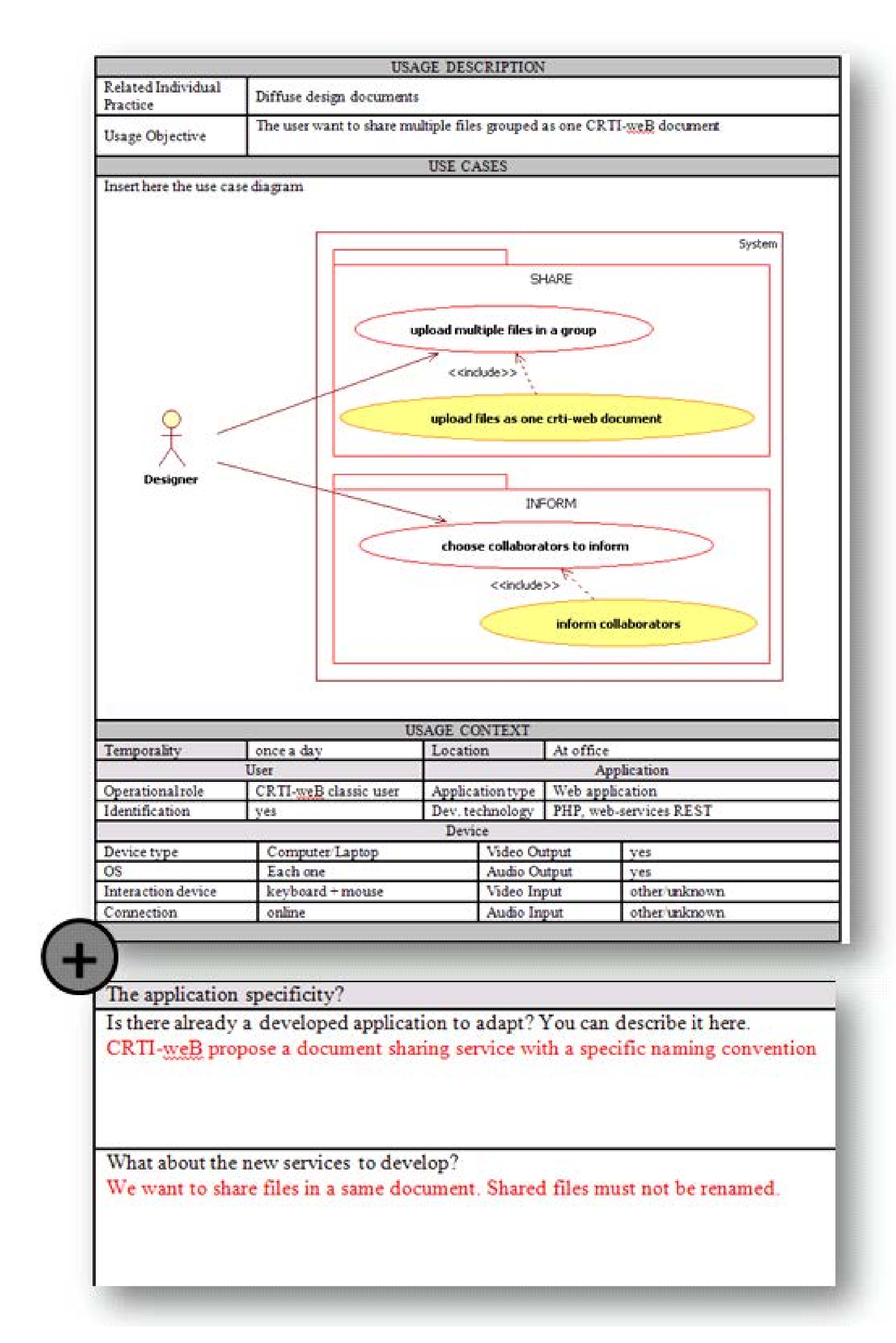
Une spécificité du premier usage concerne le format des documents échangés. Les plans sont partagés sous la forme d'un fichier .dwg (format d'enregistrement d'un plan éditable avec le logiciel autocad) et plusieurs autres fichiers souvent sous forme de pdfs : des « représentations figées » des plans (par exemple chaque étage), des spécifications techniques ou d'autres documents nécessaires à la mise en œuvre. Ici le format des fichiers n'est pas une information déterminante, mais nous verrons qu'elle peut avoir un impact non négligeable. Elle doit donc apparaitre dans notre modèle.

Vers un troisième usage

Nous avons identifié une variante de l'usage de chargement groupé, au cours duquel le format des fichiers est particulièrement important.

Il s'agit du partage de plusieurs documents au format dwg parmi lesquels il y a un fichier « maître » et des fichiers liés qui correspondent à des parties du fichier maître (fichiers identifiés comme des « xrefs » auxquels fait référence le fichier maître). Le service de partage de ces fichiers devra permettre de les grouper, mais ne devra pas les renommer afin que le lien soit conservé techniquement.

Ces informations sont typiquement le genre d'informations à consigner dans un cahier d'exigences pour garder traces des besoins identifiés. Tel que nous l'avons décrit dans le chapitre 10, notre méthode propose de les modéliser dans le « contextual use case » (Figure 127).



| USAGE DESCRIPTION | |
|---|---|
| Related Individual Practice | Diffuse design documents |
| Usage Objective | The user want to share multiple files grouped as one CRTI-weB document |

USE CASES

Insert here the use case diagram



| USAGE CONTEXT | | | |
|---|---|---|---|
| Temporality | once a day | Location | At office |
| User | | Application | |
| Operational role | CRTI-weB classic user | Application type | Web application |
| Identification | yes | Dev. technology | PHP, web-services REST |
| Device | | | |
| Device type | Computer/Laptop | Video Output | yes |
| OS | Each one | Audio Output | yes |
| Interaction device | keyboard + mouse | Video Input | other/unknown |
| Connection | online | Audio Input | other/unknown |

The application specificity?

Is there already a developed application to adapt? You can describe it here.

CRTI-weB propose a document sharing service with a specific naming convention

What about the new services to develop?

We want to share files in a same document. Shared files must not be renamed.

*Figure 127. Une partie du « contextual use case » spécifié pour le partage d'un plan dwg et ses « xrefs » (troisième usage)*

### *1.40.4 Le choix et la spécification de la solution à implémenter*

Une fois que les usages ont été identifiés et spécifiés, il s'agit de choisir ce qu'il faut implémenter et comment cela peut être réalisable. Dans un cas d'amélioration de services comme ceux de CRTI-weB, qui fait l'objet de cette première expérimentation, les choix sont fortement conditionnés par les spécificités techniques de l'existant.

La phase de spécification du service se fait à travers l'association du point de vue de concepteur avec celui des développeurs. En tant que concepteurs, nous avons proposé un diagramme de séquences « non technique » (consigné en annexes), c'est à dire ne détaillant pas l'architecture du service, mais sur la base duquel nous avons pu collaborer avec les développeurs. Cette étape doit permettre de lever deux risques majeurs :

- celui de rencontrer des problèmes techniques qui empêchent le bon développement de la solution spécifiée,
- celui de s'engager dans des développements d'envergure, longs et onéreux au lieu de privilégier une solution plus simple mais qui répondrait au besoin identifié.

Dans cette optique, il a finalement été choisi de ne jamais renommer les fichiers qui sont partagés sous la forme d'un seul document, matérialisant ainsi en un même service l'usage d'upload groupé « général » et celui plus spécifique avec les « xrefs ». C'est ainsi qu'a été proposé un service d'upload multiple sous forme de deux « sous-services », l'un pour l'envoi de documents indépendants et l'autre pour l'envoi groupé sous la forme d'un document unique.

Nous rejoignons ici le cas illustré dans le chapitre 11 décrivant la création d'un fichier zip agrégeant tous les fichiers sous un même document (Figure 128).

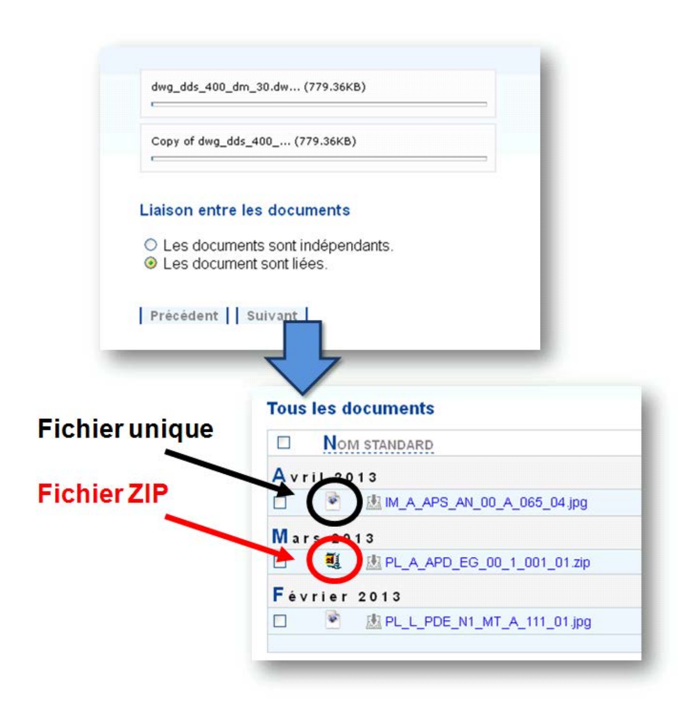



*Figure 128. Rappel : le service « d'upload multiple » développé*

### 1.40.5 Conclusion

Cette expérimentation a permis de dérouler un processus entier de développement de service informatique dans un cadre professionnel et sur la base de besoins identifiés avec des de vrais utilisateurs de la plateforme CRTI-weB. La collaboration entre notre équipe et les développeurs s'est effectuée autour de plusieurs réunions majeures dédiées à l'analyse des besoins et aux choix conceptuels. Les développements qui ont suivi ont ainsi pu être menés sans spécifications supplémentaires.

Évaluation de la pertinence

Nos premiers entretiens avec l'équipe de la société de services Kitry Consulting ont mis en avant la nécessité de maîtriser les risques comme une préoccupation importante. Les risques sont définis comme la possibilité qu'un projet ne soit pas exécuté conformément aux prévisions de dates, de coûts ou que le résultat ne satisfasse pas les besoins exprimés.

Les utilisateurs de CRTI-weB qui se projettent dans leur activité expriment des besoins très particuliers, liés au contexte de collaboration dans lequel ils utilisent l'outil (type de projet, quantité d'information échangée, autres acteurs en présence...). Dès lors, les risques qui peuvent être rencontrés sont :

- une demande impossible à réaliser techniquement, car incompatible avec le développement déjà en place, ou demandant trop de temps à l'implémentation,
- une demande incomprise par les développeurs, menant à des choix erronés qui ne satisferont pas l'utilisateur et dont la correction sera coûteuse.

C'est pourquoi la bonne gestion d'un tel projet en s'assurant que les besoins identifiés sont les bons dès les premières phases, est primordiale. Dans un secteur comme celui de la construction, nous avons vu que les besoins peuvent être très spécifiques. Ils sont dépendant du contexte métier aussi bien d'un point de vue organisationnel qu'opérationnel. Au cours de cette expérimentation, nous avons pu valider :

- l'intérêt d'une méthode permettant de prendre en compte ces points de vue
- l'intérêt d'un document traçant le suivi de cette méthode

L'approche de conception par services est de plus tout à fait appropriée à l'outil CRTI-weB lui-même ayant été défini comme un ensemble de services métiers répondant à des pratiques collectives génériques (Kubicki, et al. 2009).

Évaluation et évolution des concepts (méta-modèles) et des modèles

Cette première expérimentation a permis de valider la distinction faite entre point de vue organisationnel et opérationnel à travers les concepts de pratiques et usages. La correspondance entre pratiques et usages peut suivre plusieurs scénarii qui ont été envisagés lors de la conception du méta-modèle d'usages (MMU). Nous montrons ainsi que trois relations ont un sens dans un contexte de collaboration outillé par les services informatiques (Figure 129) :

- un usage est défini par pratique,
- plusieurs usages (ou variantes d'usage) sont envisagés pour une même pratique,
- le même usage médiatise plusieurs pratiques.

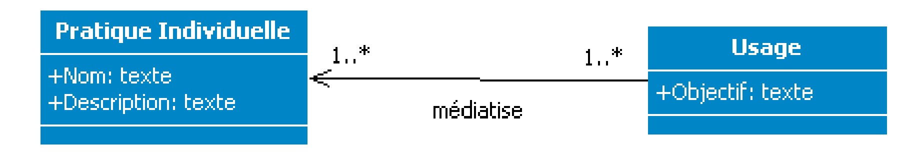



*Figure 129. Rappel : relation entre les concepts d'usage et de pratique telle que caractérisée par le MMU*

Le concept de pratique permet d'associer les requêtes de la part des utilisateurs à des besoins métiers généralisables et ainsi de les justifier auprès des développeurs. A ce propos, le diagramme de pratiques proposé (1.40.2) pour retranscrire les besoins s'est avéré compréhensible et utile.

La définition précise du contexte de l'usage avait peu d'intérêt ici car le contexte était connu des développeurs qui assurent déjà la commercialisation de l'outil, les ateliers de formation et le recueil des retours des utilisateurs. Il permet cependant, a posteriori, de garder une trace des choix effectués. Le « contextual use case» était compréhensible mais s'est avéré peu utile. Seule la partie diagramme était intéressante pour résumer de manière générale les intentions à outiller.

Nous avions proposé des diagrammes de tâches K-MAD (la conception de notre propre éditeur de tâches particulier n'a été entreprise que plus tard). La description des tâches (utilisateur et système) dans l'usage était utile dans un premier temps (première réunion) mais insuffisante pour la suite des discussions. Les besoins en termes de description étaient plus importants que ceux supportés par les arbres de tâches. C'est pourquoi nous avons introduit le diagramme de séquence qui s'est avéré le modèle le plus utile pour la discussion avec les développeurs. Cela ne remet pas en cause l'intérêt d'un arbre de tâches dans la méthode mais il s'avère plus utile lorsque l'interaction occupe une place plus importante dans l'usage.

Notons enfin que les différences entre les formats de documents (dwg, xref, pdf) manipulés auraient pu être modélisées par le concept d'objet d'interaction (cf. 1.32.2). Ce concept n'a été introduit que plus tard, mais cette expérimentation en a motivé la création.

## 1.41 Expérimentation 2 : l'automatisation du service d'upload de CRTI-weB

### *1.41.1 Introduction et déroulement*

Lors de cette deuxième expérience, nous avons collaboré avec un nouveau développeur, intégré dans l'équipe de conception pour une courte durée. Les échanges ont ainsi été favorisés et nous avons mis en place une collaboration étroite entre analystes métiers, concepteurs et développeurs (la plupart étant localisés au même endroit). Nous avons ainsi pu évaluer les capacités de l'approche à transmettre les informations nécessaires au bon développement d'un service, et ce, sans connaissance initiale du contexte (à la fois professionnel et technique). Malgré les échanges informels réguliers, nous avons menés trois séances de travail particulières ayant leurs objectifs respectifs.

La première séance (mars 2012) consistait à présenter le contexte professionnel, l'outil CRTI-weB, et les services (métiers et informatiques) qu'il propose. L'objectif général du projet a été présenté afin que le développeur puisse rechercher une technologie qui lui permette d'implémenter le futur service.

Au cours de la deuxième séance (quelques jours après) nous avons présenté les points de vue organisationnel et opérationnel au travers du cahier d'exigences (c'est-à-dire les pratiques et usages). Au terme de cette réunion, la tâche demandée au développeur a été de formaliser le point de vue fonctionnel (sous forme de diagrammes de séquences comme le préconise la méthode) en même temps qu'il développerait le service.

De la même manière que lors de la première expérimentation, la séance suivante (mai 2012) avait pour but de rendre compte des développements menés en fonction des exigences. Elle a permis d'exprimer les choix effectués en termes d'implémentation mais aussi de se questionner sur la gestion de certains cas d'erreurs. En effet, si le point de vue opérationnel exprime des scénarios idéaux d'usage, le point de vue fonctionnel doit prendre en compte toutes les possibilités à implémenter, y compris les manipulations non prévues. Au cours des développements, ces phases de précision se sont réitérées jusqu'à obtenir un service qui réponde soit fonctionnel, en plus de répondre aux exigences initiales.

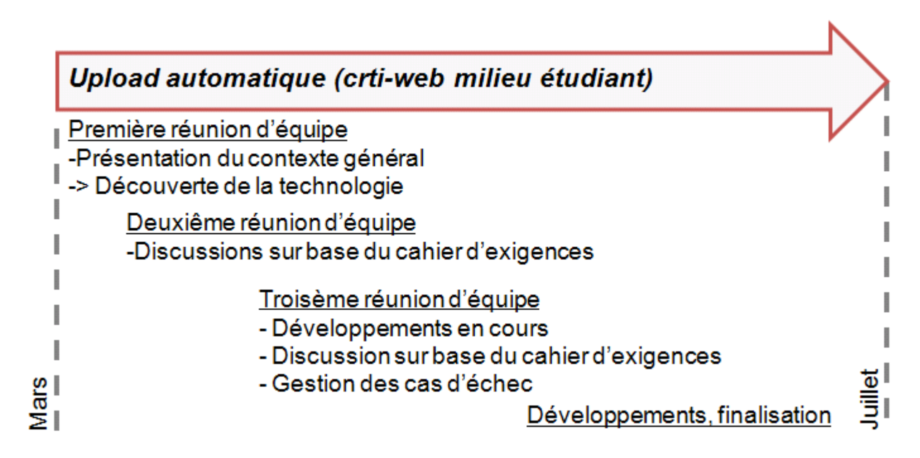



*Figure 130. Déroulement de l'expérimentation*

### *1.41.2 Un nouvel usage d'upload*

La pratique analysée dans cette expérimentation est une pratique de partage de documents proche de celles que nous avons décrites dans l'exemple précédent. Sa spécificité est qu'elle se déroule dans un milieu universitaire et non dans un contexte professionnel habituel (Figure 131). Il s'agit en effet d'un projet universitaire de conception en situation de collaboration à distance entre des étudiants de Nancy et de Liège (le Studio Digital Collaboratif (Kubicki, Guerriero, Leclercq, et al. 2009)). Les rôles dans ce projet de développement ont été attribués de la manière suivante :

- Utilisateurs : les étudiants
- Experts métiers : les encadrants académiques (ils possèdent le recul nécessaire à l'analyse du travail)
- Concepteurs : notre équipe
- Développeur : un stagiaire informaticien intégré à l'équipe pour le projet

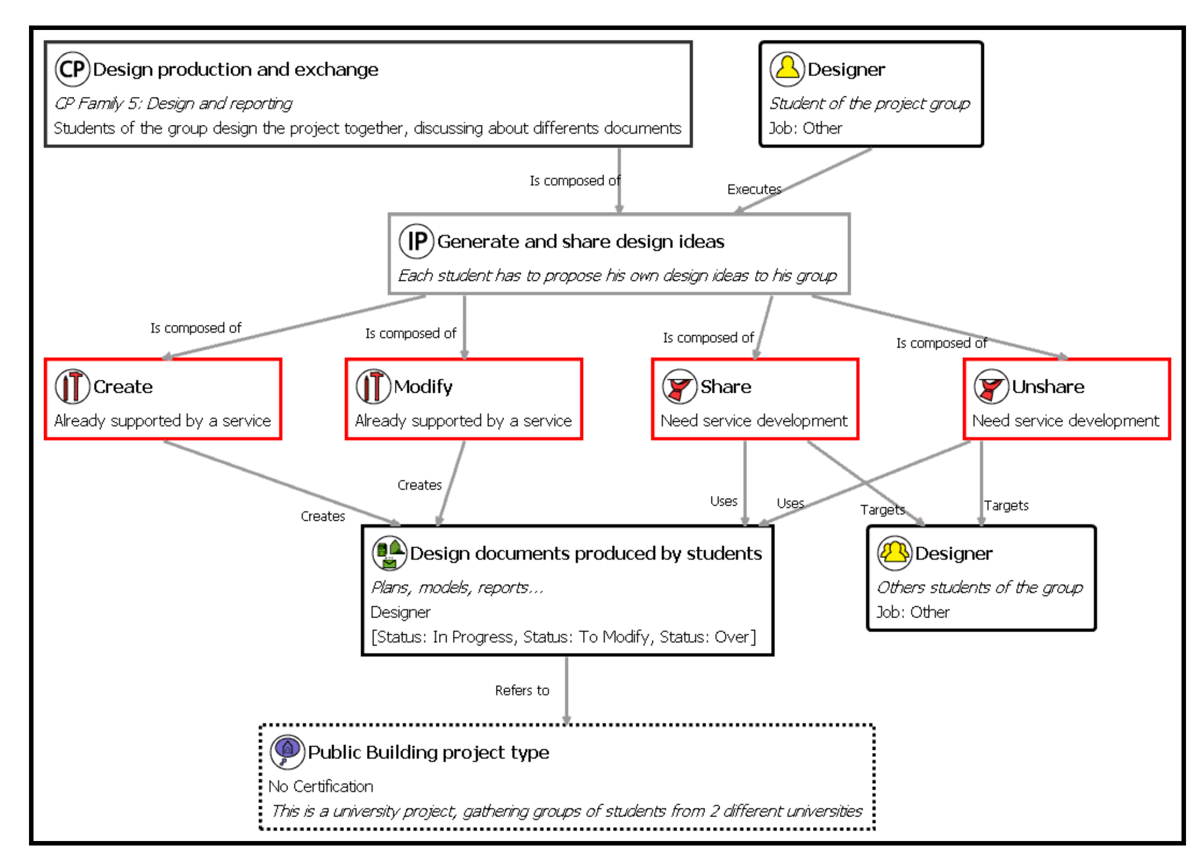



*Figure 131. Diagramme de la pratique analysée*

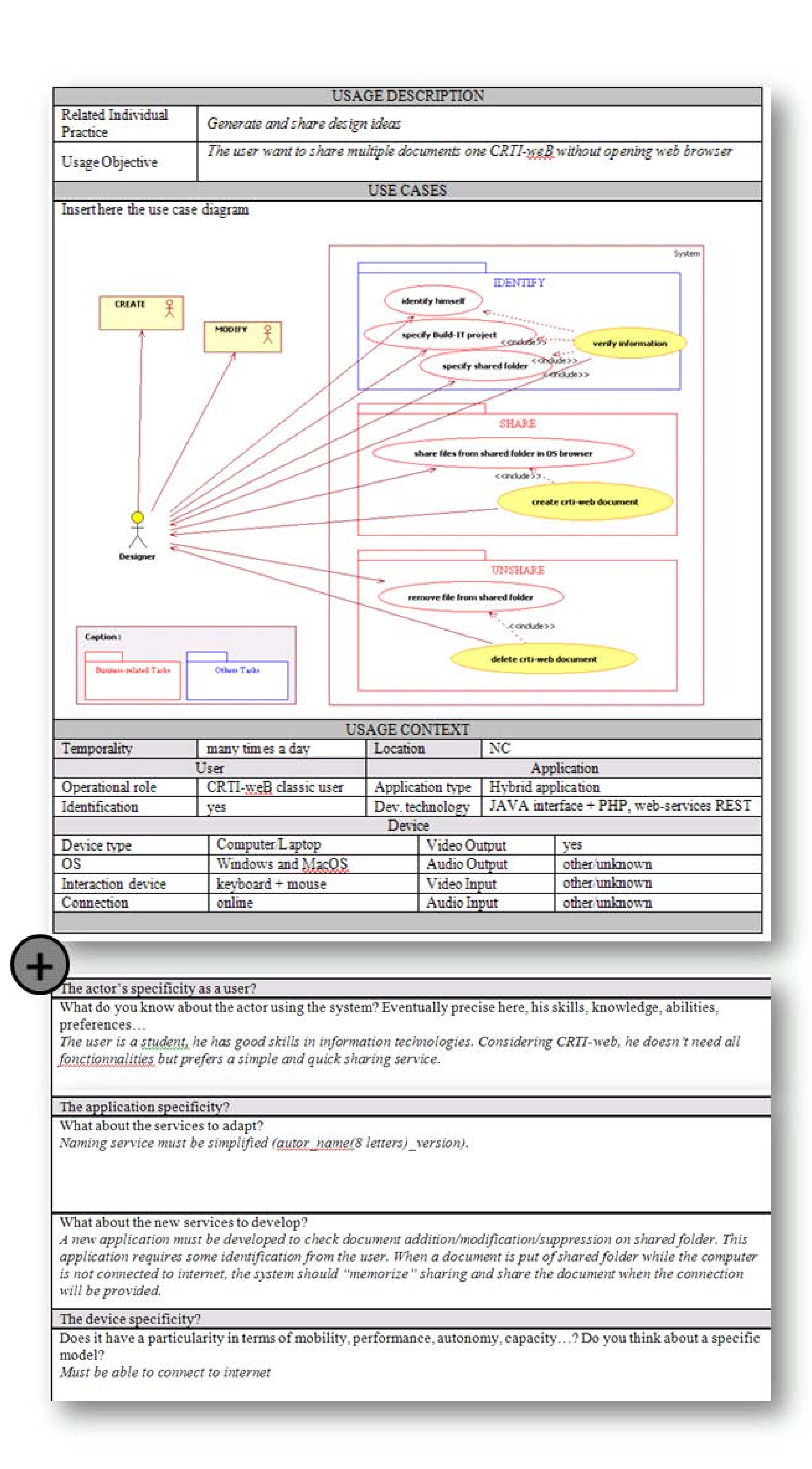
Les étudiants disposent, pour se coordonner au cours de ce projet, du service de partage de documents de CRTI-weB et du système de Bureau Virtuel présentés précédemment (voir 1.2.2). Ils peuvent également utiliser tout autre outil qu'ils jugeront utile à la gestion de leur travail.

Les étudiants sont habitués à utiliser les services de Dropbox ou Facebook pour travailler en groupe, car ils ne sont pas soumis aux restrictions du milieu professionnel (en particulier les clauses de confidentialité) et ces outils font partie de leurs usages personnels réguliers. Ainsi, nous avons pu constater dans les années antérieures qu'ils jugent souvent CRTI-weB trop complexe et, dans la plupart des cas, abandonnent son usage au profit de solutions plus simples. Des requêtes d'utilisateurs définies en termes de nouveaux services n'ont pas été formalisées comme c'était le cas pour lors de l'expérimentation précédente. Cependant, les professeurs encadrant le projet et jouant le rôle d'expert métier nous ont rapporté une analyse de ces limites actuelles de CRTI-weB : la lourdeur des étapes d'identification (login, oubli régulier des mots de passe), le temps nécessaire à la phase de nommage des documents envoyés (trop contraignant dans un tel contexte et peu adaptée à la nature des documents envoyés) et le temps d'attente lié au processus d'upload en lui-même. Pour franchir ces limites, un nouvel usage a été proposé dans le cadre de l'expérimentation. Le travail de conception était basé sur l'interprétation, à travers les concepts de la méthode PUSH, d'un usage qui nous semblait répondre à certaines de ces limites (celui de l'outil dropbox) et son adaptation à notre contexte.

Aussi, nous avons proposé un service d'upload automatique lié à CRTI-weB dont les caractéristiques principales sont :

- la surveillance d'un dossier particulier avec la détection de tout nouveau fichier ajouté dans ce dossier,
- le chargement automatique des fichiers détectés sur la plateforme avec nommage automatique (selon une convention de nommage volontairement simplifiée : initiales de l'auteur _ 8 premières lettres du nom du fichier _ indice)
- la mise à jour automatique avec incrémentation de l'indice en cas de nouvelle version d'un fichier.
- le report de l'ajout « d'actions » à un document vers un usage ultérieur qui nécessitera de repasser par l'interface web.

La figure suivante illustre le « contextual use case » qui décrit cette proposition (Figure 132). C'est au cours de cette expérimentation que nous avons introduit dans la modélisation (sous forme du package bleu) la distinction des spécifications fonctionnelles indépendantes du contexte métier mais nécessaires à la spécification du service. Ces use case « non métier » sont relatifs ici à l'identification de l'utilisateur, le choix du dossier qui sera partagé et le choix du projet CRTI-weB sur lequel les documents seront envoyés. Les deux autres packages définissent le chargement et la suppression de fichiers sur CRTI-weB à partir du dossier partagé.



| USAGE DESCRIPTION | |
|---|---|
| Related Individual Practice | *Generate and share design ideas* |
| Usage Objective | *The user want to share multiple documents one CRTI-weB without opening web browser* |

USE CASES

Insert here the use case diagram



| USAGE CONTEXT | | | |
|---|---|---|---|
| Temporality | many times a day | Location | NC |
| User | | Application | |
| Operational role | CRTI-weB classic user | Application type | Hybrid application |
| Identification | yes | Dev. technology | JAVA interface + PHP, web-services REST |

| Device | | | |
|---|---|---|---|
| Device type | Computer/Laptop | Video Output | yes |
| OS | Windows and MacOS | Audio Output | other/unknown |
| Interaction device | keyboard + mouse | Video Input | other/unknown |
| Connection | online | Audio Input | other/unknown |

**The actor's specificity as a user?**

What do you know about the actor using the system? Eventually precise here, his skills, knowledge, abilities, preferences…

*The user is a student, he has good skills in information technologies. Considering CRTI-web, he doesn't need all fonctionnalities but prefers a simple and quick sharing service.*

**The application specificity?**

What about the services to adapt?

*Naming service must be simplified (autor_name(8 letters)_version).*

What about the new services to develop?

*A new application must be developed to check document addition/modification/suppression on shared folder. This application requires some identification from the user. When a document is put of shared folder while the computer is not connected to internet, the system should "memorize" sharing and share the document when the connection will be provided.*

**The device specificity?**

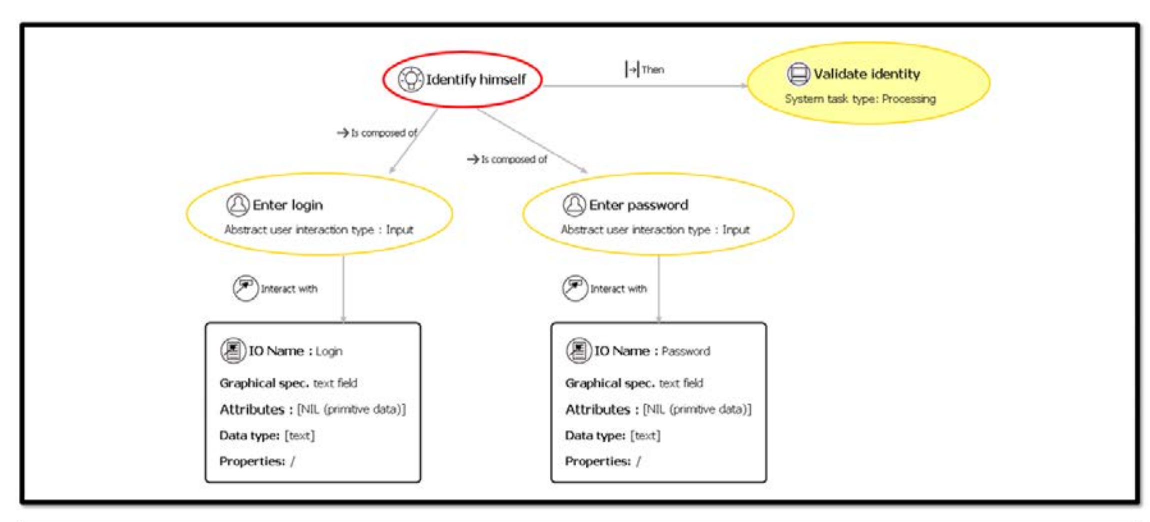
Does it have a particularity in terms of mobility, performance, autonomy, capacity…? Do you think about a specific model?

*Must be able to connect to internet*

*Figure 132. « Contextual use case » du service d'upload automatique*

Les diagrammes d'interaction suivants (Figure 133) décomposent :

- L'intention « s'identifie » en tâches « saisit le login » et « saisit le mot de passe ». Les objets d'interaction concernés (login et mot de passe) sont des données primitives. Le système devra vérifier ces données.
- l'intention « charge les fichiers à partir du dossier partagé » en « Copie », « modifie » et « renomme les fichiers dans le dossier ». Les objets avec lesquels interagissent ces tâches sont des fichiers. À partir de ces fichiers, le système devra créer un autre objet « document crti-web ».



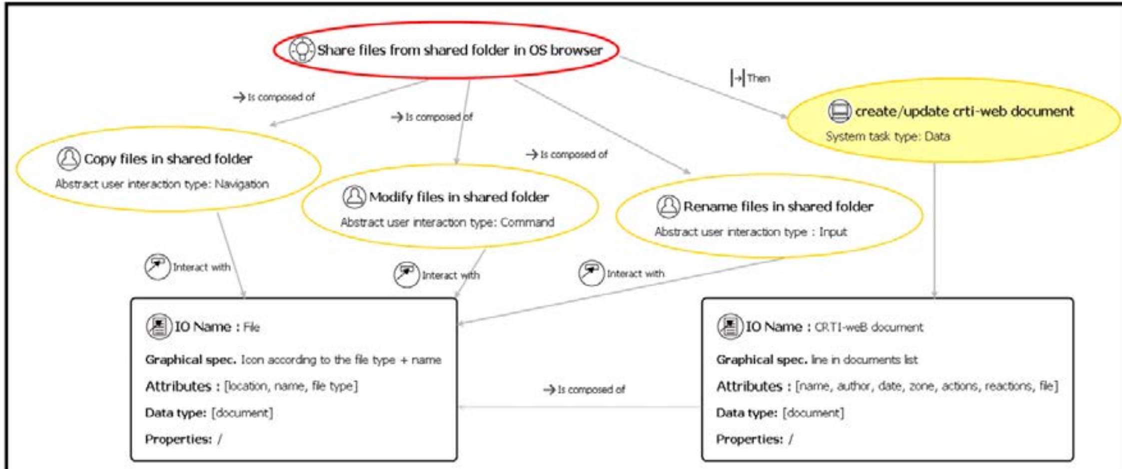




*Figure 133. Diagrammes d'interaction pour les intentions « s'identifie » et « partage les fichiers depuis un dossier partagé »*

### *1.41.3 Spécifications techniques et développements*

Une application nommée CRTI-box, composée d'une interface JAVA et d'un « daemon » (logiciel permettant de surveiller les flux de fichiers dans un dossier) a été développée pour implémenter le service conçu. Celle-ci sert de passerelle entre les services web et l'utilisateur ainsi que le dossier partagé. Elle devra être paramétrée la première fois pour effectuer les tâches d'identification (premier package) puis participera en tâche de fond à l'exécution des tâches suivantes. Le diagramme de séquence suivant illustre une partie de l'exécution du service et également un cas d'erreur. Le service web de nommage existant sur la plateforme CRTI-weB a dû être modifié, pour répondre à l'exigence de nommage automatique selon la convention « Initiales

Auteur _ 8 premières lettres du nom du fichier _ version ». Le contenu de la base de données CRTI-web n'a pas changé.

Au cours de cette expérimentation, nous avons confié l'édition des diagrammes de séquence au développeur afin d'évaluer leur utilité à rendre compte des choix de développement (auprès des concepteurs) et à les capitaliser (pour les futurs développeurs). La figure suivante en représente un extrait (Figure 134).

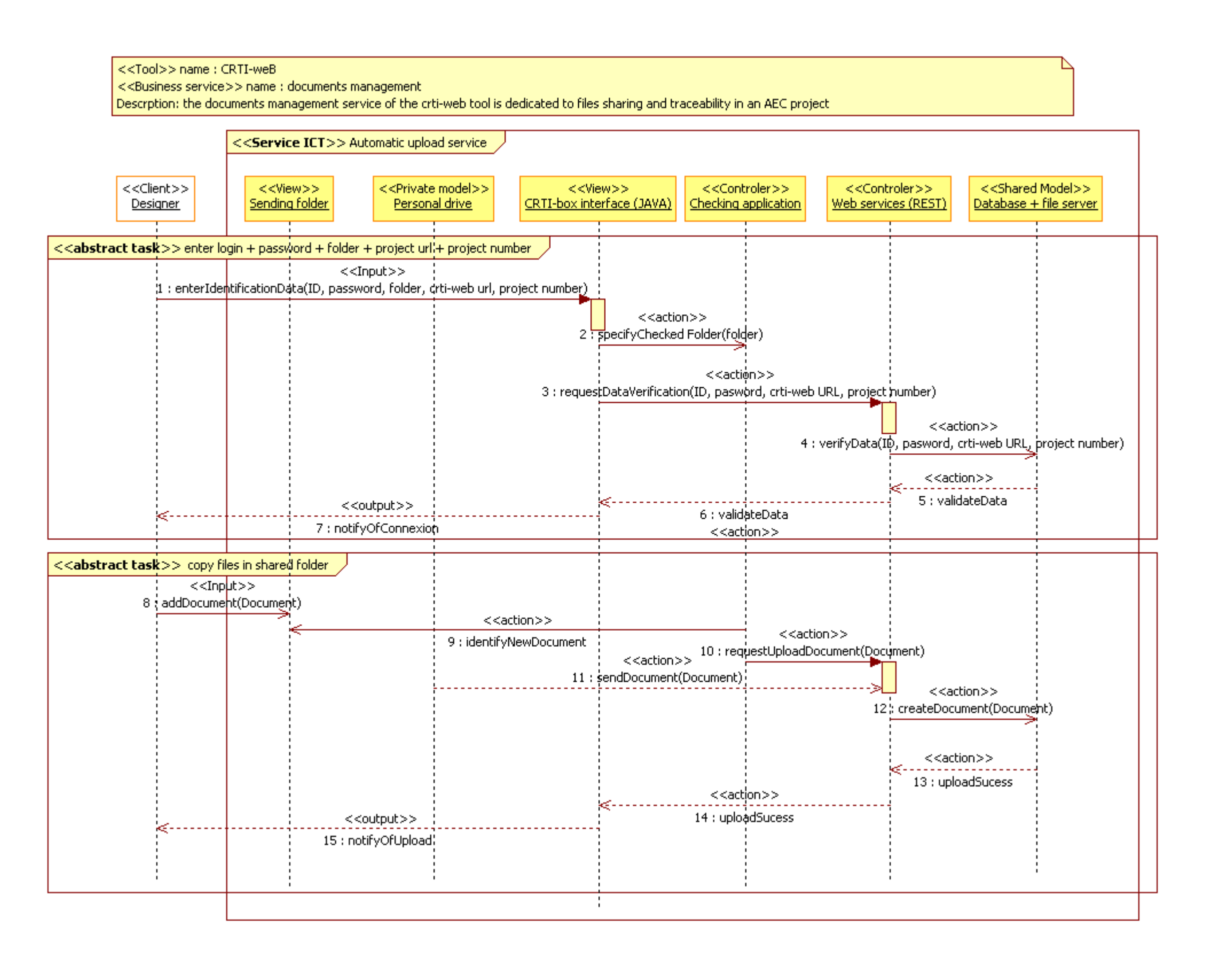



*Figure 134. Partie du diagramme de séquence spécifiant le service*

Au fil des itérations nous avons pu détailler les exigences, comme par exemple la spécification d'un message de notification pour avertir du démarrage de l'application après la phase identification. La figure suivante (Figure 135) illustre le service développé, et la façon dont il répond aux exigences (ici représentées par une version simplifiée du diagramme d'interactions).

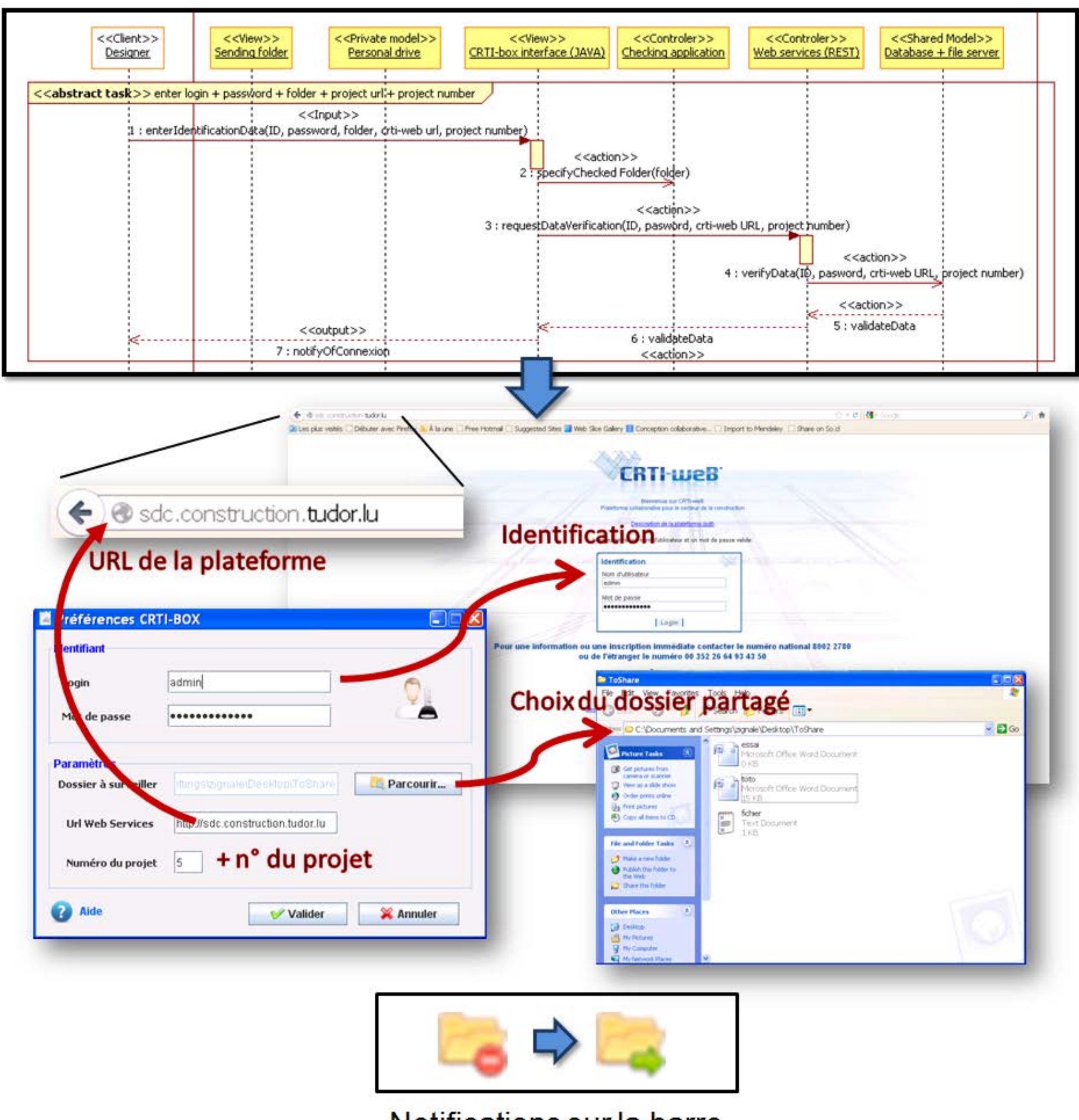



*Figure 135. Illustration du résultat obtenu au niveau de l'interface par rapport aux exigences*

### *1.41.4 Conclusion*

Cette expérimentation a permis de mener un processus de développement de service pour CRTI-weB mais dans un contexte différent de l'expérimentation n°1. Les utilisateurs étaient différents puisqu'ils étaient étudiants et non des professionnels, ce qui avait un impact sur leurs usages. De plus, le contexte de développement était également différent étant donné que le développeur ne connaissait pas du tout l'outil. La collaboration s'est également effectuée autour de réunions de synchronisation pour l'analyse des besoins et les choix conceptuels. Ce contexte a donc favorisé l'application de la méthode dans son ensemble, en prenant le temps de réaliser tous les modèles attendus. En effet, cette demande faisait partie des exigences de l'équipe vis-à-vis du développeur stagiaire, et n'aurait pas pu être attendue d'une équipe de développement soumise à

des objectifs de rentabilité, comme celle intervenant dans l'expérimentation n°1. Les développements ont donc été davantage suivis et tracés.

Évaluation de la pertinence

L'intérêt de la méthode réside pour ce cas d'étude dans la possibilité d'identifier et spécifier un nouvel usage plausible et qui illustre bien l'impact de la variation du contexte métier. Après avoir identifié les opérations essentielles à médiatiser autrement que par l'usage classique CRTI-weB (par l'interface web), un usage alternatif plus simple a été conçu. Lors de l'implication d'un nouveau membre dans un projet de développement (ici le développeur), les risques liés au manque d'information et à l'incompréhension des besoins sont plus importants. L'application de la méthode pour mener ce projet a permis de réduire ces risques en favorisant la collaboration autour des différents modèles et donc le partage de connaissances explicites.

Évaluation et évolution des concepts (méta-modèles) et des formalismes

Nous avons pu évaluer l'intérêt d'un diagramme de pratiques pour expliquer rapidement comment se positionnent les exigences métiers du point de vue organisationnel. Après la deuxième réunion de discussion autour du cahier d'exigences, le contexte métier était compris et le diagramme de pratiques servait de rappel en cas de doutes. Son caractère synthétique et ciblé sur un cas précis est un atout.

L'approche de modélisation par « contextual use case » et diagrammes d'interaction (et l'association des deux) permet d'exprimer ce qui est attendu en terme d'interaction. Choisissant de nous affranchir du formalisme de K-MAD, nous avons généré notre propre éditeur à partir du méta-modèle d'usage (MMU). Étant ainsi plus libres, nous avons pu faire évoluer le MMU pour qu'il décrive exactement les concepts que nous voulions manipuler. Pour cette expérimentation, ces modèles se sont montrés plus utiles que précédemment car il était nécessaire d'introduire une nouvelle façon d'interagir avec le système. Alors que l'expérimentation précédente était un travail portant essentiellement sur l'architecture du service, les exigences portaient ici à la fois sur le fond et la forme de celui-ci (fonctionnalités et interactions). L'utilité de modéliser les objets d'interaction n'a pas été validée ici étant donné la simplicité de ces derniers.

Enfin, les diagrammes de séquences ont été entièrement spécifiés par le développeur lui-même. Ainsi, il a pu rendre compte de sa production mais aussi des difficultés techniques auxquelles il était confronté. Les concepteurs sont bien souvent trop attachés à leurs choix conceptuels sans considérer les limites techniques. Nous concluons que la discussion autour de ces diagrammes permet de faciliter la compréhension entre ces deux mondes. C'est à partir de ces diagrammes de séquence « techniques » que nous avons pu spécifier le méta-modèle de services qui guide maintenant leur création au cours de l'application de la méthode PUSH.

## 1.42 Expérimentation 3 : spécification d'un tableau de bord

### *1.42.1 Introduction et déroulement*

Au cours de cette troisième expérimentation, la méthode n'a été appliquée que partiellement. L'enjeu portait essentiellement sur la spécification d'usages et les liens avec des pratiques et non sur le développement des services à proprement dit. C'est un travail qui a été mené par deux

personnes jouant, l'une jouant le rôle de concepteur l'autre et d'expert métier. Dans cette expérimentation, la volonté était de proposer un nouvel outil qui réponde à des pratiques existantes mais peut-être peu ou pas encore médiatisées. Ce cas relève donc plutôt d'une volonté d'innovation par la technologie, plutôt que d'une réponse à un besoin métier observé. Il nous permet d'évaluer le potentiel de notre méthode et nos méta-modèles dans un autre type de projet de conception de services.

Le tableau de bord envisagé ici est un outil de collaboration dont l'objectif est de fournir une information synthétique sur le projet de construction au cours des différentes étapes de chantier (matériaux, tâches d'exécution, malfaçons…). Il est destiné à assister le rôle de coordinateur au cours des visites de chantier régulières. L'usage que nous introduisons ici est donc relatif à un contenu et à un contexte spatio-temporel particulier. Il est également lié à un matériel particulier, à savoir une tablette tactile pouvant être géolocalisée.

L'expérimentation, beaucoup plus courte que les deux précédentes, a été composée de deux séances de travail. La première avait pour but d'initier le projet à partir de maquettages illustrant les attentes de l'expert métier. Ce dernier avait donc une idée plutôt formalisée du service à concevoir.

Lors de la deuxième séance de travail, les différents modèles préconisés par la méthode ont été composés. Le concepteur a ensuite pu formaliser le cahier d'exigences à partir des modèles et des notes prises au cours de la séance de travail.

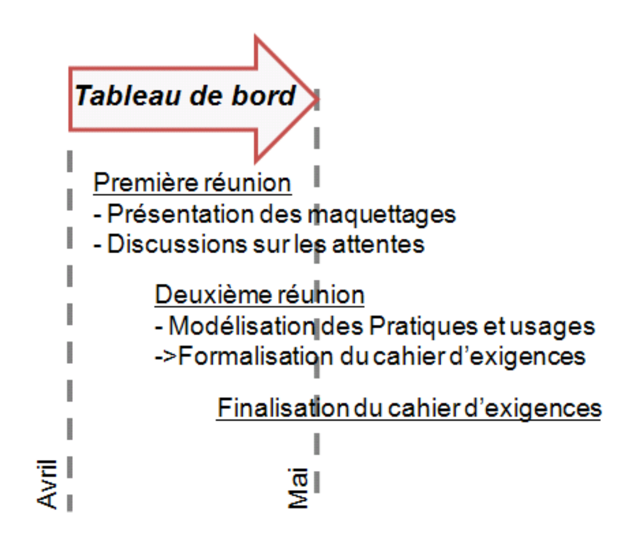



*Figure 136. Déroulement de l'expérimentation*

### *1.42.2 Analyse des besoins métier : un maquettage comme point de départ*

L'idée de départ proposée par l'expert métier était de concevoir un tableau de bord proposant un certain nombre de services liés à la localisation du coordinateur sur le chantier, pendant sa visite. À partir de ses connaissances du domaine et de ses attentes quant à l'usage d'un tel outil, l'expert métier a proposé un ensemble de maquettages qui pourraient servir de première impulsion à un projet de développement.

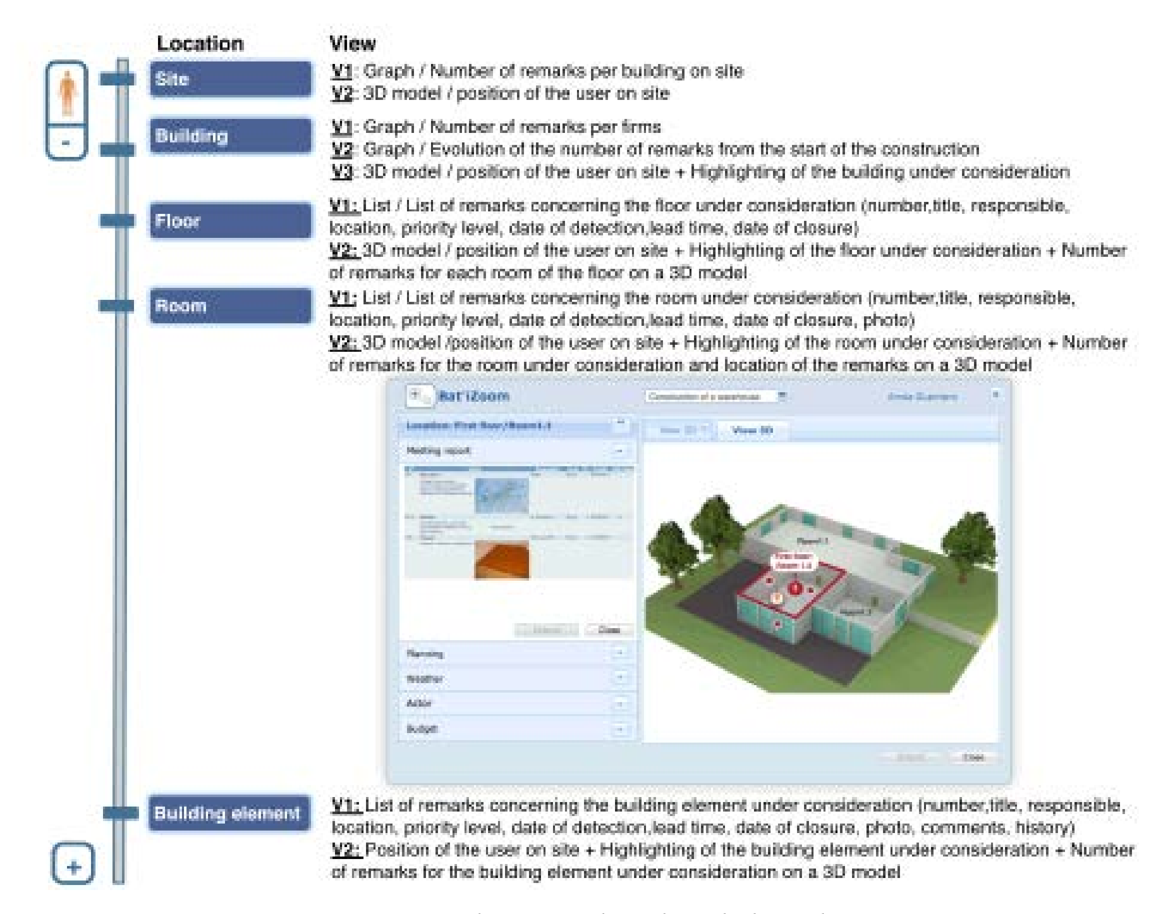



*Figure 137. Proposition d'un nouvel outil sur la base de maquettages (tiré de (Guerriero et al. 2012))*

Ces maquettages sont des vues de l'outil telles qu'elles devraient se composer en fonction de « l'échelle » considérée (ou niveau de détail), à savoir : le site, le bâtiment, l'étage, la pièce et l'ouvrage. Ils sont complétés par une description littérale de l'information à afficher dans chacune de ces vues à chacun de ces niveaux

Du point de vue organisationnel, il s'agit d'assister une pratique du coordinateur qui souhaite analyser l'avancement du chantier en visitant le bâtiment lui-même mais aussi au travers des comptes-rendus de chantier et des remarques faites sur le bâtiment. S'il identifie des défauts, il voudra en informer les entreprises concernées au travers de remarques. Nous avons pu caractériser cette pratique grâce au diagramme de pratiques. Il aura fallu compléter le méta-modèle afin de pouvoir d'ajouter des éléments de description comme par exemple les défauts.

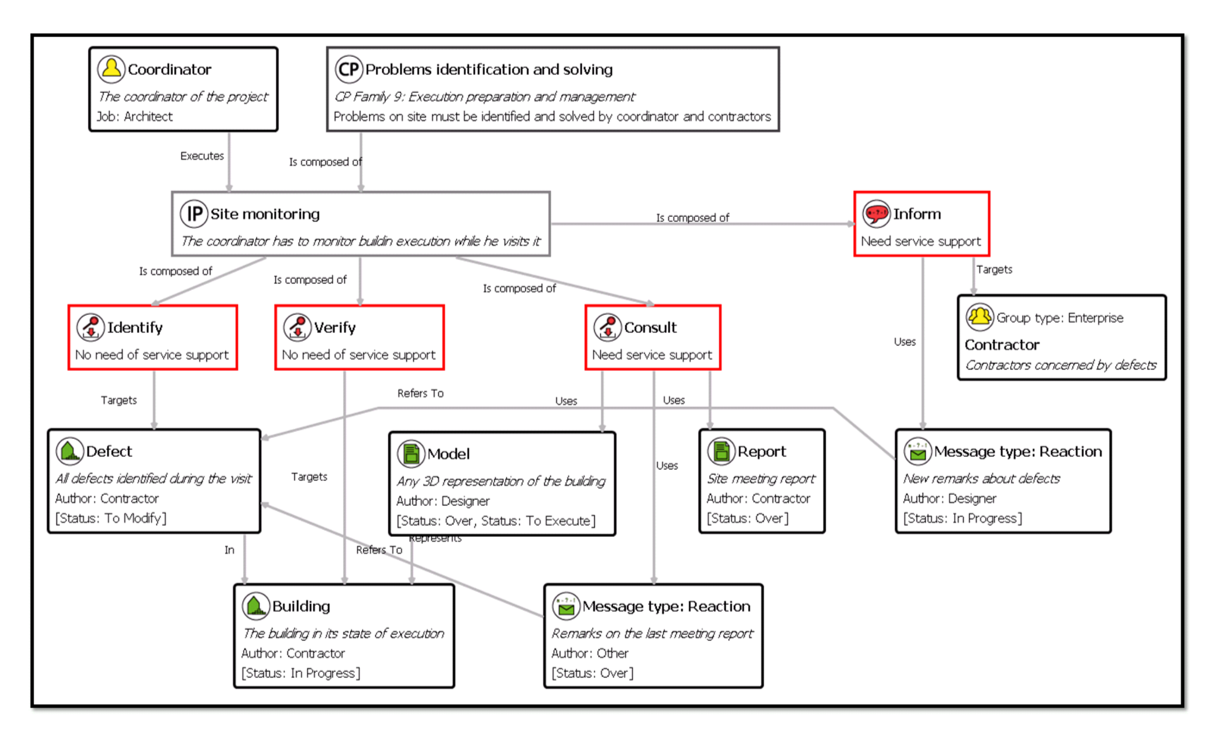
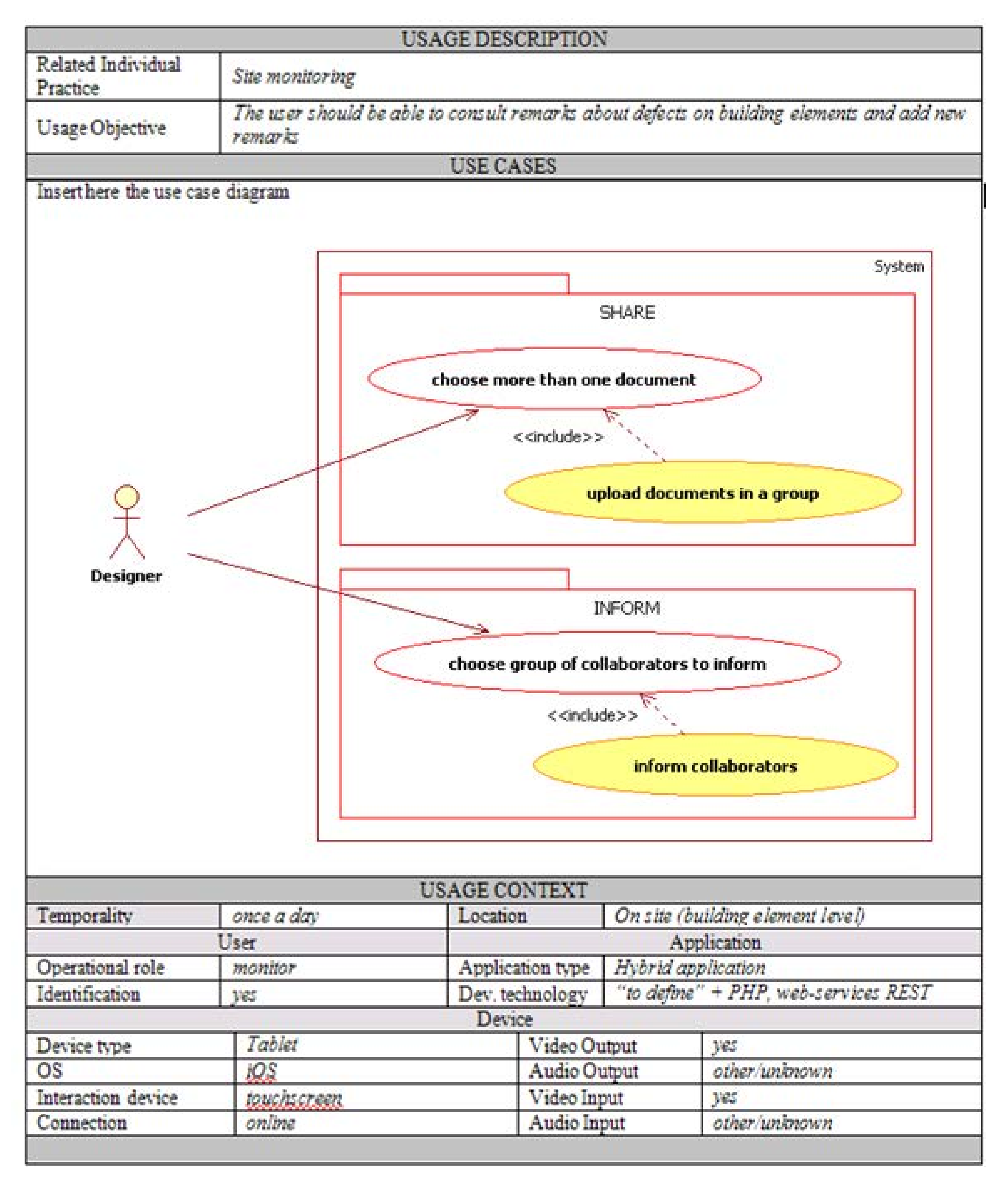



*Figure 138. Diagramme de la pratique de surveillance du chantier*

### *1.42.3 Identification et conception de l'usage, perspectives de développement*

D'un point de vue opérationnel, l'intention de l'utilisateur est dans un premier temps d'accéder aux éléments du compte-rendu et aux remarques en fonction de sa localisation dans le bâtiment, à l'échelle du site, du bâtiment, de la pièce et de l'ouvrage. Le système sera chargé d'afficher la bonne information en fonction de la localisation. Nous imaginons qu'identifier un ouvrage pour en afficher les informations relatives pourrait se faire de deux façons : en sélectionnant l'ouvrage dans le modèle 3D ou en scannant un QR Code voire une puce RFID sur l'ouvrage. On pourra définir ainsi deux variantes de l'usage. À l'échelle de l'ouvrage, il sera alors possible de recenser tout nouveau défaut remarqué et pas encore référencé afin d'en informer les entreprises.



| USAGE DESCRIPTION | |
|---|---|
| Related Individual Practice | *Site monitoring* |
| Usage Objective | *The user should be able to consult remarks about defects on building elements and add new remarks* |
| **USE CASES** | |
| Insert here the use case diagram | |



| USAGE CONTEXT | | | |
|---|---|---|---|
| Temporality | *once a day* | Location | *On site (building element level)* |
| User | | Application | |
| Operational role | *monitor* | Application type | *Hybrid application* |
| Identification | *yes* | Dev. technology | *"to define" + PHP, web-services REST* |
| Device | | | |
| Device type | *Tablet* | Video Output | *yes* |
| OS | *iOS* | Audio Output | *other/unknown* |
| Interaction device | *touchscreen* | Video Input | *yes* |
| Connection | *online* | Audio Input | *other/unknown* |

*Figure 139. Extrait du « contextual use case » pour l'usage du tableau de bord (niveau ouvrage)*

Le diagramme d'interactions suivant détaille l'intention « ajoute une nouvelle remarque ». La tâche de sélection de l'ouvrage (tâche abstraite) est ici spécifiée par un « click sur l'ouvrage dans le modèle » ou « un scan de la puce RFID » (tâches concrètes), tel que nous l'avons décrit plus haut. Les objets d'interaction sont l'ouvrage et la remarque décrivant le défaut sur cet ouvrage. L'ouvrage est un objet du modèle qu'on caractérise comme objet BIM, la remarque est à la fois élément du modèle (objet BIM) et élément du rapport de chantier (document).

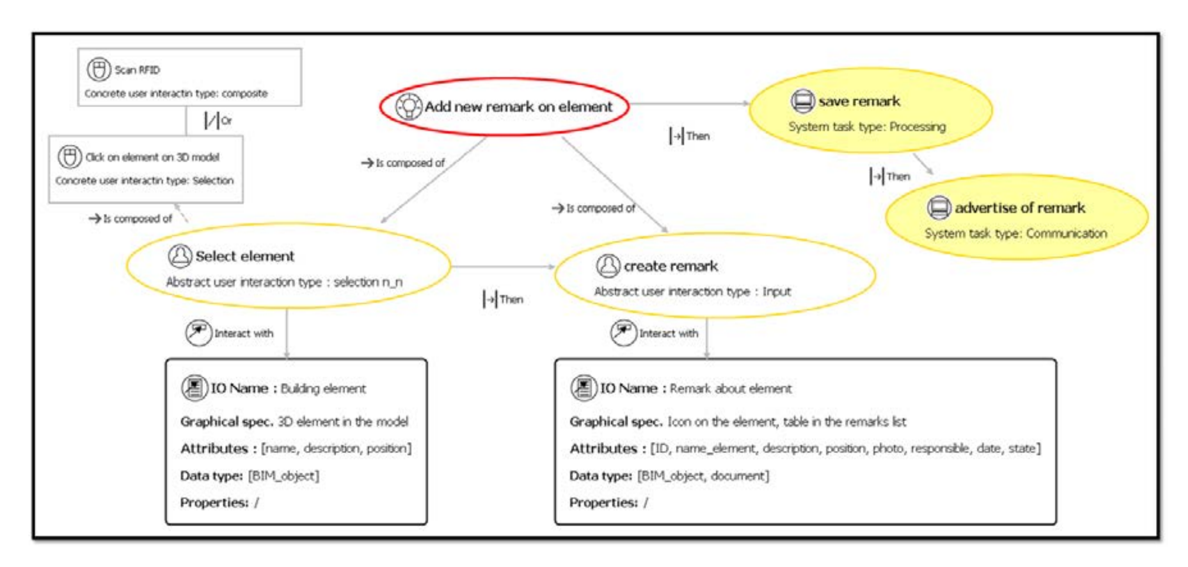



*Figure 140. Diagramme d'interactions pour l'ajout de remarques*

Recommandations et pistes pour le développement du service

Les remarques sur le chantier sont les informations clés de cet usage et sont, comme nous venons de le dire, des éléments du rapport de chantier. Sachant que le service métier « comptes-rendus » de CRTI-weB propose déjà des services informatiques pour la gestion de ces rapports et des remarques, nous pourrions les intégrer dans l'implémentation du nouveau service. Ce service métier de « surveillance de chantier » devra aussi contenir des services relatifs au BIM.

### 1.42.4 Conclusion

Cette expérimentation nous a permis d'appliquer la méthode de manière différente et pour un type de service différent. L'idée innovante initiale (un tableau de bord utilisant les remarques des comptes-rendus CRTI-weB) a pu être en partie concrétisée en réponse à des pratiques métiers modélisées grâce à notre méthode et par une définition des usages liés.

Évaluation de la pertinence

Lorsqu'un utilisateur ou un expert métier propose un service très précis à travers un maquettage, le risque majeur est qu'il ne perçoive pas les limites du réalisable dans le contexte technologique dans lequel il se trouve. Il ne peut pas non plus évaluer les coûts et le temps requis pour mener le projet de développement à bien. D'un point de vue collaboratif, une proposition sur simple maquettage ne pourra donc pas suffire auprès d'une société de services car, avant tout début de projet :

- il faudra vérifier qu'il y ait de réels besoins métiers à outiller,
- il faudra évaluer l'impact d'une telle proposition en termes de développements.

La méthode PUSH permet, par la modélisation, d'ancrer une volonté d'innovation dans un processus de conception justifié. Elle encourage ainsi l'amorce d'un projet de développement auprès de ses acteurs.

Évaluation et évolution des concepts

Nous avons dû faire évoluer les méta-modèles afin de pouvoir décrire les situations de la manière la plus fidèle. Nous avons par exemple ajouté l'objet « défaut » nécessaire ici dans la caractérisation de la pratique. Par extension, en nous questionnant sur les éléments relatifs au chantier qui n'étaient pas encore caractérisés, nous avons également ajouté les engins de chantier, les matériaux et les échantillons. L'approche dirigée par les modèles, et en particulier la génération de notre propre interface de modélisation, permet donc une évolution qui sera sûrement nécessaire au fil d'autres applications de la méthode.

Dans cette expérimentation, nous avons pu exploiter le concept d'objet d'interaction et décrire ainsi les ouvrages du bâtiment et les remarques du compte-rendu comme de réels « éléments métiers médiatisés ». Cela nous permet de définir un usage au plus proche des pratiques métiers identifiées en amont.

Évaluation des formalismes

Nous avons constaté que lorsqu'une situation devient plus complexe (avec de multiples opérations, des liens entre les artefacts…) le diagramme de pratiques perd en lisibilité. Cela fait ressortir l'importance de traiter les exigences au cas par cas sans définir des processus trop complets. Les correspondances graphiques entre modèles permettent de tenir un discours cohérent avec un collaborateur qui n'a pas l'habitude de manipuler autant de modèles. Les diagrammes d'interactions gagnent également à être divisés en plusieurs entités pour une meilleure lisibilité.

## 1.43 Conclusion : apports des expérimentations

### *1.43.1 Résumé*

La première expérimentation nous a permis de mettre en évidence l'importance de la spécification des besoins métiers et les différences entre les usages qui peuvent en découler. Elle a montré l'importance de la collaboration entre experts métiers, concepteurs et développeurs autour de modèles appropriés. Elle a également permis de valider les concepts de pratique, usage et service comme concepts clés des étapes de conception d'un service.

La deuxième expérimentation a montré un autre cas d'amélioration de service en se concentrant sur la simplification de l'interaction au niveau de l'usage. Elle a fait ressortir l'importance :

- de décrire le contexte métier et l'utilité de la modélisation de pratique,
- de décrire le contexte technique, ce qui est attendu et l'utilité de la modélisation de l'usage,
- de décrire ce qui est proposé et l'utilité de la modélisation du service.

La troisième expérimentation a validé l'intérêt de la méthode en tant que moyen de formaliser un processus de conception à partir d'une idée de service (décrite par un maquettage). Elle a permis de préciser les concepts utilisés et d'introduire les objets d'interaction en tant qu'élément de description de l'information métier médiatisée.

Transmettre l'information : le rôle des modèles

Un projet de conception de services est un projet collaboratif en lui-même, ce qui implique l'intervention de plusieurs points de vue sur le processus. Nous avons associé à ces points de vue plusieurs modèles qui permettent de les exprimer (Figure 141). Les expérimentations ont permis d'observer le rôle de ces modèles.

Les modèles de pratiques permettent aux experts métier de décrire de manière simple une situation observée. Les utilisateurs eux-mêmes ne peuvent être impliqués dans une description complète de leur travail. Cependant, lorsqu'il s'agit de proposer ou d'améliorer un service pour les assister, ils peuvent mettre en avant des problèmes particuliers auxquels ils sont confrontés. Un modèle de pratique devra permettre de formaliser de telles situations particulières tout en les conceptualisant afin de pouvoir associer ce modèle à d'autres cas observés. L'expert métier crée ce modèle et le communique aux concepteurs de services.

Les concepteurs font état du contexte de l'usage à travers le « contextual use case ». Celui-ci est présenté aux experts métier pour une description générale de la solution qui sera implémentée et permet une première phase de validation. Dans un deuxième temps, il est complété de diagrammes d'interaction (voire de maquettages) et soumis aux développeurs pour ouvrir la discussion sur les aspects techniques. Un premier ensemble de diagrammes de séquences « non technique » (sans détail de l'architecture du service) présentant un scénario idéal peut également être proposé par les concepteurs.

Les développeurs mènent la production du service et produisent de manière ponctuelle une version détaillée des diagrammes afin de supporter la discussion. Il s'agit alors d'échanger sur les alternatives à proposer en cas de limites techniques ou sur la gestion des cas d'erreur.

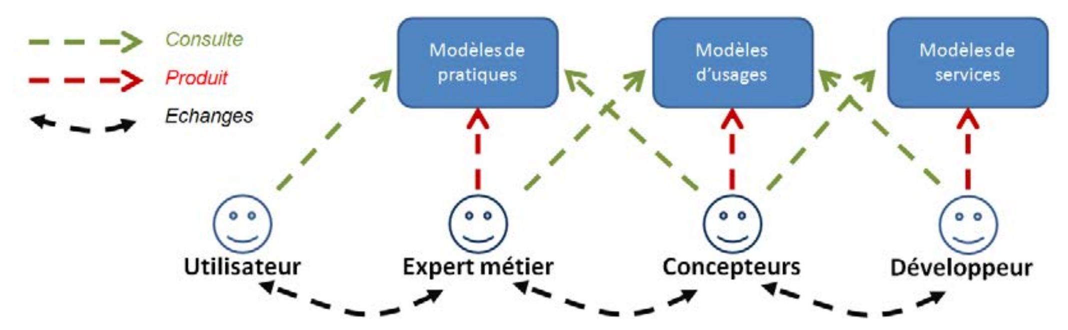



*Figure 141. Collaboration pour la conception de services autour de la production de modèles*

L'importance des formalismes

Notre choix s'est porté sur une association des modèles graphiques et des modèles textuels pour une expression des points de vue alliant synthèse et précision. Les diagrammes sont adaptés pour décrire les comportements en présentant de manière synthétique les acteurs et les entités manipulées que ce soit au niveau des pratiques métiers comme au niveau de l'usage ou encore pour la description du service. Leur lecture est simple si peu d'éléments les composent. Afin de limiter la détérioration de la lecture, il faudra donc privilégier la décomposition des cas trop

complexes en cas simples. L'avantage d'une approche graphique est de pouvoir lier visuellement les modèles entre eux, et représenter ainsi les liens conceptuels entre les points de vue. Si les diagrammes de cas d'utilisation ou de tâches font parties des modèles adoptés par la communauté, l'utilisation des arbres hiérarchiques pour la description du métier diffère quant à elle des habituels diagrammes de processus comme BPMN. Instanciés à partir d'un méta-modèle de pratiques construit sur la base d'une étude du contexte d'un projet AIC, ils permettent cependant une description à la fois spécifique et pertinente du métier.

La description textuelle qui complète les modèles dans le cahier d'exigences permet quant à elle de détailler certains éléments sur lesquels il sera important de s'attarder. Les éléments à décrire sont précisés et formatés dans des tableaux. Leur édition est guidée par choix multiple ou totalement libre. Les choix multiples doivent permettre de couvrir tous les cas de figure ou à défaut d'être complétés si besoin. L'édition libre présente l'intérêt de pouvoir s'exprimer librement sans contrainte. Il peut néanmoins être difficile dans certains cas de savoir quoi répondre.

Vis-à-vis des expérimentations menées, nous pensons qu'un modèle pertinent n'est pas forcément un modèle complet (dans lequel on aura utilisé et défini tout les concepts) mais un modèle concis, qui reprendra l'information strictement nécessaire à la conception du service final.

Résumé des réponses des expérimentations

Le début de ce chapitre (cf. § 1.39.3, Tableau 24) évoquait nos attentes en termes de validation vis-à-vis des expérimentations. Nous pouvons à présent conclure sur les apports réels de ces expérimentations (Tableau 25). Les flèches rouges symbolisent l'évolution de la méthode (notamment des concepts et formalismes utilisés) grâce aux expérimentations. Les symboles « + » montrent que les expérimentations se complètent dans la validation de la méthode, chacune apportant un élément différent.

Tableau 25. Apports des différentes expérimentations

| Apports… | Upload multiple | | Upload Automatique | | Tableau de bord |
|---|---|---|---|---|---|
| **cf. méthode** | En favorisant le transfert des exigences, la méthode permet de lever des risques | + | La méthode permet d'intégrer les développeurs dans le projet et favorise le dialogue avec les concepteurs | + | La méthode permet aux experts métiers et aux concepteurs de s'accorder sur un usage à spécifier en fonction des besoins métier |
| **cf. concepts** | Les descriptions des pratiques et usages sont pertinentes mais ne suffisent pas à décrire le service | → | La composition des trois points de vue forme un modèle de développement complet | + | Les concepts peuvent être améliorés et complétés de manière régulière pour raffiner l'analyse |
| **cf. modèles** | Le diagramme de pratique est utile mais les diagrammes relatifs aux usages restent insuffisants sans plus de précisions techniques | → | Le passage d'un modèle à l'autre est explicite et favorise le dialogue entre concepteurs et développeurs | + | Le passage d'un modèle à l'autre est explicite et favorise le dialogue entre expert métier et concepteur de service |
| **cf. adaptation** | Les utilisateurs privilégient l'upload multiple et en font l'usage pour lequel il a été développé | | Peu de tests on été effectués, ils valident le fonctionnement du service mais pas son adaptation | | Aucun développement n'a été effectué |

### *1.43.2 Les trois « push »*

Les trois expérimentations menées nous ont permis d'identifier trois impulsions possibles pour le développement d'un service adapté : c'est ce que nous définissons comme les trois « push » de la méthode.

Le push métier

Le push métier, c'est l'impulsion issue d'un besoin d'assistance à des pratiques exécutées dans un contexte métier particulier. Cette assistance devra permettre d'atteindre les objectifs métiers identifiés en assurant la satisfaction des utilisateurs lors de la création ou l'adaptation de services. La création de CRTI-weB ainsi que les phases ultérieures de développement comme celle constituant notre première expérimentation sont issues d'une telle impulsion.

Le push technique

Le push technique permet d'assister une pratique de manière différente par une réponse technologique alternative à des solutions existantes. Il s'agit de définir un autre usage qui sera perçu comme plus simple par les utilisateurs tout en assurant l'exécution de leur pratique métier. La définition de CRTI-box (deuxième expérimentation) est issue d'une telle impulsion basée sur l'étude et l'adaptation des services de DropBox.

Le push innovation

Le push innovation est basé sur la création de nouveaux contextes, que ce soit d'un point de vue métier ou technologique. En d'autres termes, on cherchera à faire émerger de nouvelles pratiques ou de nouveaux usages par la conception de services. La proposition d'un Tableau de bord sur tablette tactile et profitant de la géolocalisation pour assister la coordination d'un chantier était typiquement synonyme d'une volonté d'innovation. Le risque dans un tel cas de figure est cependant de ne pas mesurer les limites qu'il faudra franchir pour développer le service voulu.

# Conclusion et perspectives

Le travail décrit dans cette thèse a porté sur deux points essentiels : l'analyse métier et la conception de services informatiques. Nous avons mis en avant les relations qui existaient entre ces deux points à travers la modélisation et afin de répondre à une problématique : la conception de services collaboratifs adaptés aux pratiques d'un projet AIC.

## Concevoir une méthode de conception : recul sur une approche orientée « design science »

Le cycle itératif composé des phases « connaissance, suggestion, développement, évaluation et conclusion » a été suivi dans la conception de la méthode proposée (Figure 142). Les paragraphes suivants en résument succinctement le contenu.

La connaissance multidisciplinaire a été construite autour de trois axes de recherche principaux :

- des études initiales sur le contexte coopératif d'un projet AIC,
- du projet de conception des services de la plateforme CRTI-weB,
- et d'un état de l'art sur les méthodes de conception logicielle selon plusieurs approches.

À partir de ces connaissances, plusieurs processus de conception et développement ont été menés et évalués au cours de trois expérimentations. Nous avons ainsi proposé :

- une modélisation des pratiques métiers qui complète les études antérieures sur le contexte coopératif,
- et l'intégration de cette modélisation dans un processus de conception de services, inspiré des méthodes analysées et appliqué pour l'amélioration des services de CRTI-weB.

Les expérimentations ont également permis de prendre un certain recul sur le déroulement d'une méthode de conception de services en concluant sur divers points :

- il est important de supporter chaque point de vue apporté dans un processus de conception en distinguant les concepts manipulés par chacun d'eux, et en facilitant le transfert de l'information essentielle entre ces points de vue,

- il est important d'utiliser des langages et formalismes qui soient adaptés aux différents points de vue, en considérant les langages existants et les habitudes de chacun.

Enfin, il a évidemment été question à travers les expérimentations de valider la réponse à la problématique initiale, à savoir l'adaptation de services proposés aux besoins métiers.

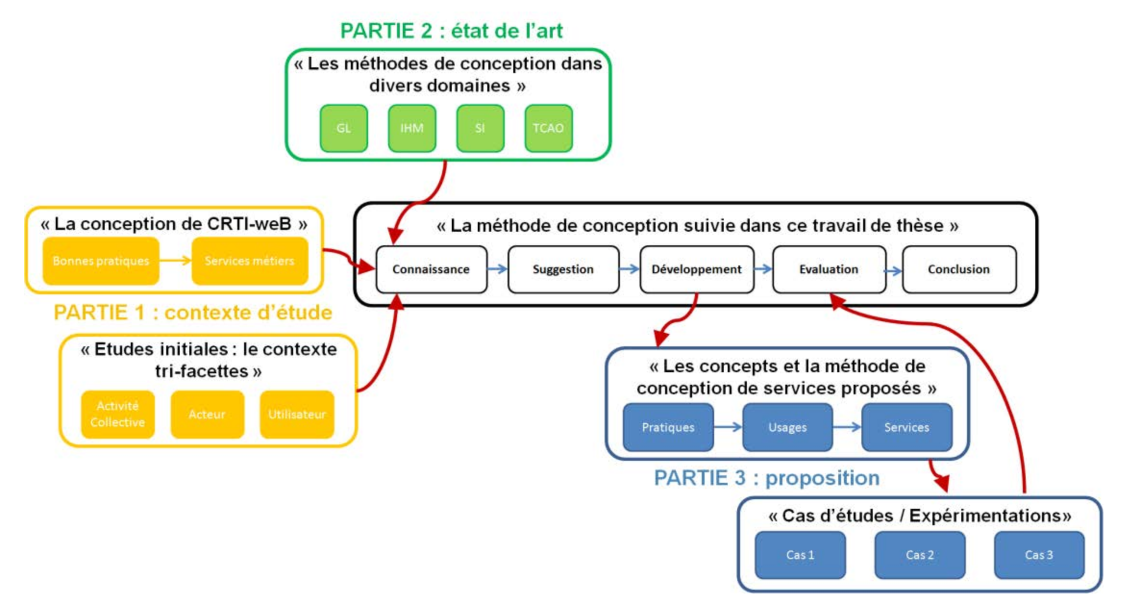



*Figure 142. Rappel : Démarche suivie dans notre recherche basée sur la conception*

L'approche fait preuve cependant de quelques lacunes sur plusieurs points. C'est ce dont nous discuterons dans les paragraphes suivants.

## Les limites de l'approche

Les premières limites sont d'ordre théorique et sont relatives à l'exploration des divers champs de recherche. Malgré une étude approfondie des méthodes de conception de service, le manque d'expertise initiale dans ce domaine pourra se refléter dans l'état de l'art proposé. Notre approche non-experte a permis de prendre du recul par rapport aux références du domaine et de théoriser celles-ci pour en extraire une méthodologie générale. Cependant, on pourra reprocher à cette étude de ne pas entrer dans les détails, notamment techniques et relatifs au développement.

Il est important de rappeler que l'approche proposée n'a pas la prétention de vouloir générer des services adaptés à la manière des approches MDA. La manière de typer les tâches et les données que nous proposons est une première étape dans cette optique mais cette réflexion n'a pas été éprouvée. Notre approche fournit par contre les éléments nécessaires en vue d'améliorer la spécification de services dans le cadre d'un projet de conception logicielle « classique ». L'attention a notamment été portée sur la gestion et la coordination des différents points de vue au cours du processus de conception de services.

Trois expérimentations ont été menées afin de valider la proposition formulée. Cependant, on pourra se demander dans quelle mesure les méta-modèles ne seraient pas perfectibles et améliorables par leur instanciation dans d'autres cas d'études. Il serait intéressant de les tester de

manière étendue jusqu'à ce qu'ils se stabilisent pour ainsi valider leur pertinence de manière représentative pour l'ensemble des projets. Il faudra pour ça survoler un panel d'expérimentations plus étendu. Étendre les expérimentations permettra également d'évaluer l'appropriation d'une telle approche par des équipes différentes. Actuellement, nous n'avons pu soumettre l'approche à d'autres experts métiers ou concepteurs que ceux impliqués dans le projet CRTI-weB et les services parallèles. Le projet CRTI-box a introduit un nouveau développeur mais cela ne permet pas de valider l'application de la méthode de manière étendue.

On pourra remarquer également que, au contraire des approches d'ingénierie des exigences classiques, cette méthode ne prend pas en compte de manière explicite les exigences non fonctionnelles. Comparée aux méthodes de conception d'IHM, elle ne permet pas non plus d'inclure des critères d'évaluation ergonomique.

## Perspectives

Les champs de recherche couverts ouvrent des perspectives multiples à ce travail, notamment par rapport aux limites identifiées ci-dessus.

Il a déjà été évoqué la nécessité d'étendre l'application de cette méthode à de plus nombreux cas d'études pour en évaluer la pertinence à la fois théorique et pratique. Pour cela, il sera indispensable d'améliorer l'édition des différents modèles par les acteurs concernés. Plusieurs pistes sont à suivre :

- intégrer les éditeurs de modèles dans un seul outil qui permettra de les faire évoluer de manière homogène, sans avoir à gérer de nombreuses applications,
- l'automatisation des transformations de modèles, notamment lorsque c'est possible, par la génération automatique des éléments d'un modèle à partir du modèle précédent. Cela réduira le temps de modélisation et assurera la bonne correspondance entre les modèles.
- La génération automatisée et incrémentale des versions du cahier d'exigences pour chaque cas traité.

Au fur et à mesure des applications, nous pourrons bénéficier d'un répertoire de pratiques prédéfinies et outillées sur laquelle nous pourrons nous appuyer pour d'autres développements. L'objectif à long terme est de pouvoir associer de plus en plus d'analyses métier à des besoins déjà traités pour réutiliser, ne serait-ce qu'en partie, le travail de conception déjà effectué.

Par la séparation des points de vue et la « transformation » des modèles d'un point de vue à un autre, cette méthodologie s'inspire des approches MDA dont nous avons présenté les caractéristiques précédemment (voir 1.11.2). Nous n'avons pas exploré, pour la méthode PUSH, les possibilités de transformation automatisée. Cependant, assister le passage d'un modèle à l'autre reste une opportunité, grâce à l'approche par composition de méta-modèles. Une perspective serait d'automatiser la création du squelette d'un modèle à partir du modèle précédent. Par exemple, la base du diagramme de cas d'utilisations (l'utilisateur, le système et les packages) pourrait être créée à partir du diagramme de pratiques, ne laissant au concepteur qu'à spécifier les cas d'utilisation pour chaque package. La création du diagramme d'interactions pourrait être assistée de la même façon. Dans une dernière étape, la méthode PUSH pourra ainsi servir d'input à une approche générative comme Caméléon (Calvary et al. 2003). Il s'agira alors de réduire la phase 3 de spécification du service, par la génération automatique de code avec une intervention limitée des développeurs.

D’un autre coté, la prise en compte des exigences non fonctionnelles et leur intégration dans le processus d’édition des différents modèles est également à envisager pour parvenir à des spécifications qui soient contractuellement pertinentes. Il en va de même pour les critères ergonomiques qui seront à intégrer pour proposer des solutions satisfaisantes en termes d’utilisabilité.

La méthode trouvera aussi sa place dans la tendance actuelle de gestion de projets de construction basée sur la maquette numérique BIM. Comme abordée dans le chapitre 7 (section 1.21), l’approche BIM a deux composantes : l’utilisation partagée d’outils de maquettes numériques (centralisées ou décentralisées) d’une part, et la mise en place de pratiques collectives maîtrisées et améliorées d’autre part. L’effort de standardisation entrepris depuis une quinzaine d’années a conduit à la normalisation d’un standard de description des ouvrages (IFC, ISO 16739:2013). Cependant, la description des processus (sous la forme d’Information Delivery Manuals) en reste aujourd’hui à ses débuts, et les chercheurs reconnaissent une certaine lourdeur de ces modèles et une difficulté à définir des processus reproductibles. Notre proposition pourrait donc certainement apporter un nouveau point de vue à ces travaux.

Nous avons envisagé l’applicabilité de cette méthode à un autre secteur que celui du projet AIC. En effet, d’autres contextes de projet de conception collaborative pourraient être analysés grâce aux outils proposés afin de concevoir des services adaptés à leurs pratiques (comme par exemple les pratiques d’ingénierie civile ou encore de conception automobile). Cependant, il sera nécessaire d’évaluer dans quelle mesure les méta-modèles et formalismes devront évoluer afin de permettre la description de ces contextes différents. Leur structure est générique, on pourra ainsi réutiliser les concepts introduits (acteurs, artafects, outils…). Cependant, il faudra adapter l’instanciation de ces concepts en fonction du domaine analysé, notamment à travers les types énumérés (type d’acteur, type de document, type d’outil…).

# Table des matières

# Bibliographie

Aalst, W., 1998. The application of Petri nets to workflow management. *Journal of circuits, systems, and computers*, pp.1–53.

Abras, C. & Maloney-Krichmar, D., 2004. User-centered design. *Bainbridge, W. Encyclopedia of*, pp.1–14.

Aguilar-Savén, R.S., 2004. Business process modelling: Review and framework. *International Journal of Production Economics*, 90(2), pp.129–149.

Andersen, P., Carstensen, P.H. & Nielsen, M., 2000. Dimensions of coordination. In M. Schoop & C. Quix, eds. *LAP 2000. The fifth International Workshop on the Language-Action Perspective on Communication Modelling*. Aachen, Germany, pp. 41–60.

Anderson, R., 1994. Representations and requirements: The value of ethnography in system design. *Human-computer interaction*.

Armand, J., 1997. *140 séquences pour mener une opération de construction: Des études préalables à l'achèvement de l'ouvrage: actions techniques et démarches administratives*, Le Moniteur.

Arsanjani, A., 2004. Service-oriented modeling and architecture. *IBM developer works*, (January), pp.1–15.

Avignon, L. et al., 2002. *Architectures de Systèmes d'Information* Livre blanc,

Baida, Z., Gordijn, J & Omelayenko, B., 2004. A shared service terminology for online service provisioning. In M. Janssen, H. Sol, & R. Wagenaar, eds. *Proceedings of the 6th international Conference on Electroinic Commerce*. pp. 1–10.

Bain, D., Chalon, R. & David, BT, 2009. Réalité Mixte, de la conception à l'implémentation: méthodologie de transformation de modélisations IRVO vers la plateforme WComp. In *Actes des quatrièmes journées de l'AFRV*. Lyon, France.

Baron, M. et al., 2006. K-MADe: un environnement pour le noyau du modèle de description de l'activité. In *IHM "06 Proceedings of the 18th International Conferenceof the Association Francophone d"Interaction Homme-Machine*. pp. 287 – 288.

Bass, L, Faneuf, R. & Little, R., 1992. A metamodel for the runtime architecture of an interactive system. *ACM SIGCHI*

Bass, L, Clements, P. & Kazman, R., 2003. *Software Architecture in Practice*, Addison-Wesley Professional.

Beck, K, 2003. *Test-Driven Development by Example*, Addison-Wesley Professional.

Beck, K, 2006. *Extreme Programming Explained*,

Bennett, K. & Layzell, P., 2000. Service-based software: the future for flexible software. In *Software Engineering Conference, 2000. APSEC 2000. Proceedings. Seventh Asia-Pacific*. pp. 214–221.

Berard, O. & Karlshoej, J., 2012. Information delivery manuals to integrate building product information into design. *ITcon Electronic Journal of Information Technology in Construction*, 17, pp.64–74.

Bignon, J.-C., Halin, G. & Kubicki, S., 2009. Qualité et processus de mise en oeuvre du bâtiment. In *Ramau 5 - La qualité architecturale, Acteurs et enjeux*. pp. 127–142.

Bispo, C.P. et al., 2010. Applying a model-driven process for a collaborative service-oriented architecture. *The 2010 14th International Conference on Computer Supported Cooperative Work in Design*, pp.378–383.

Bjekovic, M. & Kubicki, S, 2011. Service quality description—a business perspective. In *2011 Federated Conference on Computer Science and Information Systems*. pp. 513–520.

Björk, B.C., 2002. A formalised model of the information and materials handling activities in the construction process. *Construction Innovation: Information, Process, Management*, 2(3), pp.133–149.

Björk, B.C., 1999. Information Technology in Construction–domain definition and research issues. *International Journal of Computer Integrated Design And Construction*, 1(1), pp.3–16.

Bobroff, J. et al., 1993. Les formes d'organisation des projets. In *Ecosip, Pilotages de projet et entreprises: diversité et convergences*. Paris, France, pp. 35–39.

Bodart, F. & Vanderdonckt, J., 1996. Widget standardisation through abstract interaction objects. In *In Advances in Applied Ergonomics*. USA publishing, pp. 300–305.

Boehm, B., 1988. A spiral model of software development and enhancement. *Computer*, 21(5), pp.61–72.

Booch, G. et al., 2007. *Object Oriented Analysis & Design with Application*, Addison-Wesley Professional.

Booth, P.A., 1989. *An Introduction To Human-Computer Interaction*, Lawrence Erlbaum Associates.

Bose, D., 2002. *Component-Based Development - Application in software engineering*,

Bouattour, M., 2005. *Assistance à la conception coopérative fondée sur la sémantique des ouvrages. Application au domaine du bois.* Nancy Université.

Boucher, E., 2009. *Software as a Service - Quelle est la maturité de ce marché et les possibilités d'utilisation par les entreprises?* HEC Paris.

Bourguin, G. & Derycke, A., 2005. Systèmes Interactifs en Co-évolution Réflexions sur les apports de la Théorie de l'Activité au support des Pratiques Collectives Distribuées. *Revue d'Interaction Homme-Machine*, 6(1), pp.1–31.

Brown, A., Conallen, J. & Tropeano, D., 2005. Introduction: Models, modeling, and model-driven architecture (mda). In S. Beydeda, M. Book, & V. Gruhn, eds. *Model-driven Software Development*. Springer, pp. 1–16.

Bézivin, J. & Kurtev, I., 2005. Model-based technology integration with the technical space concept. In *Metainformatics Symposium*. Springer Verlag.

Calvary, G. et al., 2003. A unifying reference framework for multi-target user interfaces. *Interacting with Computers*, 15(3), pp.289–308.

Cardoso, J. et al., 2004. Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3), pp.281–308.

Cesare, S. & Lycett, M., 2003. Business modelling with UML: distilling directions for future research. In *Enterprise information systems IV*. Kluwer Academic Publishers, pp. 153–162.

Chang, Y., Lim, Y. & Stolterman, E., 2008. Personas: from theory to practices. In *Proceedings of the 5th Nordic conference …*. pp. 439–442.

Chen, G., 2009. *Architectural Practice Simplified: A Survival Guide and Checklists for Building Construction and Site Improvements as Well as Tips on Architecture, Building Design, Construction and Project Management*, Denver, Colorado: Outskirts Press.

Cherbakov, L. et al., 2005. Impact of service orientation at the business level. *IBM Systems Journal*, 44(4), pp.653–668.

Clot, Y., 2007. De l'analyse des pratiques au développement des métiers. *Éducation et didactique*, 1(1).

Cockburn, A., 2000. *Writing Effective Use Cases*, Addison-Wesley Professional..

Cockburn, A. & Jones, S., 1995. Four principles of groupware design. *Interacting with Computers*, 7(2), pp.195–210.

Cohn, M., 2003. *User stories applied: For agile software development*,

Constantine, L, 2001. Structure and style in use cases for user interface design. In *Object Modeling and User Interface Design: Designing Interactive Systems*. Addison-Wesley Professional.

Constantine, L., 2002. Process Agility and Software Usability: Toward Lightweight Usage-Centered Design. *Development*, 1(June 2001), pp.1–10.

Constantine, L., 2006. Users, Roles, and Personas. In *The Persona Lifecycle*. Morgan-Kaufmann.

Constantine, L. & Lockwood, L. a. D., 2003. Usage-centered software engineering: an agile approach to integrating users, user interfaces, and usability into software engineering practice. In *Software Engineering*. IEEE Computer Society, pp. 746–747.

Constantine, L. & Windl, H., 2003. Usage-Centered Design: Scalability and Integration with Software Engineering. *Human-Computer Interaction: theory and Practice*.

Constantine, L, 2003. Canonical abstract prototypes for abstract visual and interaction design. In J. Jorge, N. Jardim Nunes, & J. Falcão e Cunha, eds. *Information Systems - Design, Specification, and Verification*. Springer-Verlag Berlin Heidelberg 2003, pp. 1–15.

Constantine, L, 1998. Rapid abstract prototyping. In *Software Development*. pp. 1–9.

Cooper, A., 1996. Goal-directed design.

Crawford, C., Bate, G. & Cherbakov, L., 2005. Toward an on demand service-oriented architecture. *IBM Systems*, 44(1), pp.81–107.

Crnkovic, I., Larsson, S. & Chaudron, M., 2005. Component-based development process and component lifecycle. *Computing and Information Technology - CIT 13*, 4, pp.321–327.

Curtis, B., 1992. Process modeling. *Communications of the ACM*, 35(9), pp.75–90.

David, B, 2001. IHM pour les collecticiels. *Réseaux et systèmes répartis, Hermès, Paris*, 13, pp.169–206.

Davis, F., 1989. Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS quarterly*, 13(3), pp.319–340.

Davis, W., 1999. The requirements specification.

Delgado, A., 2010. Tool support for Service Oriented development from Business Processes. In *2nd International Workshop on Model-Driven Service Engineering (MOSE'10)*.

Delotte, O., 2006. *CoCSys: une approche basée sur la construction d'un modèle comportemental pour la conception de systèmes collaboratifs mobiles*. Ecole Centrale de Lyon.

Dewan, P., 2001. An integrated approach to designing and evaluating collaborative applications and infrastructures. In *Computer Supported Cooperative Work (CSCW)*. Kluwer Academic Publishers, pp. 75–111.

Dey, A., 2001. *Understanding and using context*,

Dibbern, J. et al., 2009. *Design, implementation, and evaluation of an ICT-supported collaboration methodology for distributed requirements determination*,

Dubois, E., Gray, P. & Nigay, L., 2002. ASUR ++: a Design Notation for Mobile Mixed Systems. In *Proceedings of the 4th International Symposium on Mobile Human-Computer Interaction*. Springer-Verlag London, UK, pp. 123–139.

Dupuy-Chessa, S, 2011. *Modélisation en Interaction Homme-Machine et en Système d'Information: A la croisée des chemins*. Laboratoire d'Informatique de Grenoble.

Dupuy-Chessa, S. et al., 2010. A software engineering method for the design of mixed reality systems. In *The Engineering Of Mixed Reality Systems*. Springer London, pp. 313–334.

Eastman, C. & Sacks, R., 2010. Introducing a new methodology to develop the information delivery manual for AEC projects. In *CIB W78 2010*. Cairo, Egypt, pp. 16–18.

Eastman, CM & Jeong, Y., 2010. Exchange model and exchange object concepts for implementation of national BIM standards. *Journal of Computing in Civil Engineering*, 27, pp.25–34.

Ellis, C. & Wainer, J., 1994. A conceptual model of groupware. *Proceedings of the 1994 ACM conference on Computer supported cooperative work - CSCW '94*, pp.79–88.

Eloranta, L. & Kallio, E., 2006. *A Notation Evaluation of BPMN and UML Activity Diagrams*,

Emig, C. & Weisser, J., 2006. Development of SOA-based software systems-an evolutionary programming approach. In *Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006)*. IEEE Computer Society.

Endrei, M. et al., 2004. *Patterns: Service-Oriented Architecture and Web Services*, IBM Corp., International Technical Support Organization.

Engeström, Y., 1987. *Learning by Expanding. An Activity-theoretical approach to developmental research*,

Favre, J.-M., 2004. Towards a Basic Theory to Model Model Driven Engineering. In *Workshop on Software Model Engineering, WISME 2004*. Lisboa, Portugal.

Favre, L. & Pereira, C., 2007. Improving MDA-based Process Quality through Refactoring Patterns. In *Proc. of the 1st International Workshop on Software Patterns and Quality*. Citeseer.

Fernandez, A., 2010. *Les nouveaux tableaux de bord des managers* Eyrolles, ed.,

Fielding, R., 2000. *Architectural styles and the design of network-based software architectures*. University of California, Irvine.

Frankel, D. et al., 2003. The Zachman Framework and the OMG's Model Driven Architecture. *Business Process*, pp.1–14.

Front, A., Rieu, D. & Giraudin, J.-P., 2009. Une vision sur les problématiques actuelles de la recherche en Systèmes d 'Information. *sigma.imag.fr*, p.27.

Gabay, J. & Gabay, D., 2008. *UML 2 Analyse et conception - Mise en oeuvre guidée avec études de cas*, Paris: Dunod.

Gifford, B. & Enyedy, N., 1999. Activity centered design: Towards a theoretical framework for CSCL. In *Proceedings of the 1999 conference on Computer support for collaborative learning*.

Giraldo, W., Molina, A. & Collazos, C., 2008. A Model Based Approach for GUI development in groupware systems. In R. . Briggs, ed. *CRIWG*. Springer-Verlag Berlin Heidelberg 2008, pp. 324–339.

Godart, C et al., 2001. Implicit or explicit coordination of virtual teams in building design. In *Computer-Aided Architectural Design Research In Asia-CAADRIA'01*.

Gordijn, J., Akkermans, H. & Van Vliet, H., 2000. Business modelling is not process modelling. *Conceptual Modeling for E-Business and the Web*, pp.40–51.

Gottschalk, K., Graham, S. & Kreger, H., 2002. Introduction to web services architecture. *IBM systems*, 41(2).

Greenberg, S., 1991. Personalizable groupware: Accommodating individual roles and group differences. In *Proceedings of the second conference on European Conference on Computer-Supported Cooperative Work*. Kluwer Academic Publishers, pp. 17–32.

Greenberg, S., 2006. Toolkits and interface creativity. *Multimedia Tools and Applications*, 32(2), pp.139–159.

Greer, D. & Hamon, Y., 2011. Agile software development. In *Software - Practice and experience*. pp. 943–944.

Gregor, S., 2009. Building theory in the sciences of the artificial. *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology - DESRIST '09*.

Grinter, R., Herbsleb, J. & Perry, D., 1999. The geography of coordination: dealing with distance in R&D work. In *Proceedings of the international ACM SIGGROUP conference on Supporting group work*.

Grudin, J., 1994. Computer-supported cooperative work: History and focus. *Computer*, 27(5), pp.19–26.

Grudin, J., 1988. Why CSCW applications fail: problems in the design and evaluation of organization of organizational interfaces. *Proceedings of the 1988 ACM conference on Computer-supported cooperative work - CSCW '88*, pp.85–93.

Guerriero, A., 2009. *La représentation de la confiance dans l'activité collective - Application à la coordination de l'activité de chantier de construction*. Université Henri Poincaré - Nancy 2.

Guerriero, A., Zignale, D. & Halin, G, 2012. A Zoomable Location-Based Dashboard for Construction Management. In *CDVE2012*. Osaka, Japan.

Guibert, O., 2007. *Cours d'Analyse et Conception des Systèmes d'Information (d'Outils et Modèles pour le Génie Logiciel)*,

Göransson, B., Lif, M. & Gulliksen, J., 2003. Usability Design-Extending Rational Unified Process with a New Discipline. In *Interactive Systems: Design, Specification and Verification, 10th International Work-shop, DSV-IS 2003*. Madeira Island, Portugal: Springer-Verlag Berlin Heidelberg 2003, pp. 316–330.

Hanrot, S., 2005. Une évaluation de la qualité architecturale relative aux points de vue des acteurs. *Cahiers RAMAU*, 5, pp.pp. 111–126.

Hanser, D., 2003. *Proposition d'un modèle d'auto-coordination en situation de conception, application au domaine du bâtiment.* Institut National Polytechnique de Lorraine

He, H., 2003. What is service-oriented architecture. *Publicação eletrônica em*, 30, pp.1–5.

Henri, F. & Lundgren-Cayrol, K., 2001. *Apprentissage collaboratif à distance: pour comprendre et concevoir les environnements d'apprentissage virtuels*,

Heuvel, W., 2009. Software service engineering: Tenets and challenges. In *PESOS'09*. Vancouver, Canada: IEEE, pp. 26–33.

Hevner, A. et al., 2004. Design science in information systems research. *MIS quarterly*, 28(1), pp.75–105.

Hietanen, J., 2006. *IAI Official IFC Model View Definition Format*,.

Hodgson, G., 2003. Capitalism, complexity, and inequality. *Journal of Economic Issues*, 37(2), pp.471–478.

Huang, E. & Mynatt, E., 2006. Secrets to success and fatal flaws: The design of large-display groupware. *IEEE Computer Graphics and Applications*, 26(1), pp.37–45.

Hussain, S. et al., 2010. Mapping of SOA and RUP: DOA as Case Study. *Journal of computing*, 2(1), pp.104–109.

Ibrahim, M. & Krawczyk, R., 2003. The level of knowledge of CAD objects within the building information model. In *Association for Computer-Aided Design in Architecture*. pp. 172–177.

Idoughi, D. & Kolski, C., 2009. Vers un développement orienté services des applications interactives dans le domaine de la logistique, étude de cas. In *Workshop International IEEE Logistique et Transport*. Sousse, Tunisie.

Jackson, M. & Zave, P., 1995. Deriving specifications from requirements: an example. In *Proceedings of the 17th international conference on Software engineering*. ACM, pp. 15–24.

Kaindl, H. & Jezek, R., 2002. From usage scenarios to user interface elements in a few steps. In C. Kolski & Jean Vanderdonckt, eds. *Computer-Aided Design of User Interfaces III*. Valenciennes, France, pp. 91–102.

Kalnins, A. et al., 2010. From requirements to code in a model driven way. In D. Grundspenkis, ed. *Advances in Databases and Information Systems*. Springer-Verlag Berlin Heidelberg 2010, pp. 161–168.

Kaner, C. & Falk, J., 1999. *Testing Computer Software*,

Katsumata, M., 2007. Personalized groupware service for collaborative communities. In *IADIS International Conference Applied Computing*. pp. 510–514.

Kavakli, E., 2002. Goal-Oriented Requirements Engineering: A Unifying Framework. *Requirements Engineering*, 6(4), pp.237–251.

Kohlborn, T. et al., 2009. Identification and Analysis of Business and Software Services—A Consolidated Approach. *IEEE Transactions on Services Computing*, 2(1), pp.50–64.

Kohlmann, F. & Alt, R., 2007. Business-driven service modeling-a methodological approach from the finance industry. In L. Maciaszek & W. Abramovic, eds. *Business Process and Services Computing (BPSC'07)*. Leipzig, Germany, pp. 180–193.

Kroll, P. & Kruchten, P, 2003. *The rational unified process made easy: a practitioner's guide to the RUP*, Addison-Wesley Professional.

Kruchten, P., 2001. What Is the Rational Unified Process? , p.11.

Kubicki, S. et al., 2006. Assistance to building construction coordination – towards a multi-view cooperative platform. *ITcon Vol. 11*, 11(December 2005), pp.565–586.

Kubicki, S., 2006. *Assister la coordination flexible de l'activité de construction de bâtiments. Une approche par les modèles pour la proposition d'outils de visualisation du contexte de coopération.* Université Henri Poincaré - Nancy 2.

Kubicki, S., Guerriero, A., Leclercq, P., et al., 2009. Cooperative design studios in education Lessons learnt from two experiments. In *13th Congress of Iberoamerican Society of Digital Graphics. From Modern to Digital: the Challenges of a Transition.* , Brazil.

Kubicki, S., Dubois, E., et al., 2009. Towards a Sustainable Services Innovation in the Construction Sector. *Advanced Information Systems Engineering*, pp.319–333.

Kubicki, S., Guerriero, A. & Johannsen, L., 2009. A service-based innovation process for improving cooperative practices in AEC. *ITcon Electronic Journal of Information Technology in Construction*, XX(January), pp.1–21.

Kurtev, I., Bézivin, J & Akşit, M., 2002. Technological spaces: An initial appraisal. , p.6.

Kuutti, K., 1996. Activity theory as a potential framework for human-computer interaction research. *Context and consciousness: Activity theory and human-computer interaction*, pp.9–22.

Kvan, T., 2000. Collaborative design: what is it? *Automation in Construction*, 9(4), pp.409–415.

Laaroussi, A., 2007. *Assister la conduite de la conception en architecture: vers un système d'information orienté pilotage des processus*. Institut National Polytechnique de Lorraine.

Lamsweerde, A. Van, 2001. Goal-oriented requirements engineering: A guided tour. *Requirements Engineering, 2001.*

Lamsweerde, A. Van, 2003. Goal-oriented requirements engineering: From system objectives to UML models to precise software specifications. *... Conference on Software Engineering*, (May), pp.1–81.

Lapouchnian, A., 2005. *Goal-oriented requirements engineering: An overview of the current research*,

Laurillau, Y., 2002. *Conception et réalisation logicielles pour les collecticiels centrées sur l'activité de groupe: le modèle et la plate-forme Clover*.

Laurillau, Y. & Nigay, L., 2002. Clover architecture for groupware. *Proceedings of the 2002 ACM conference on Computer supported cooperative work - CSCW '02*, p.236.

Leontiev, A., 1978. *Activity, consciousness, and personality*, Prentice Hall.

List, B., 2006. An evaluation of conceptual business process modelling languages. In *Proceedings of the 2006 ACM symposium on Applied computing*. Dijon, France: ACM.

Loniewski, G., Armesto, A. & Insfran, E., 2011. Incorporating Model-Driven Techniques into Requirements Engineering for the Service-Oriented Development Process. In *Engineering Methods in the Service-Oriented Context*. IFIP International Federation for Information Processing, pp. 102–107.

Loucopoulos, P. & Karakostas, V., 1995. *System Requirements Engineering*, McGraw-Hill, Inc. New York, NY, USA ©1995.

Lovelock, C. & Wirtz, J., 1981. *Services marketing*, Prentice-Hall, Englewood Cliffs, New Jersey.

Lu, S. & Paris, C., 1999. Towards the automatic generation of task models. *Engineering for Human-Computer Interaction*.

Lucquiaud, V., 2005. Proposition d'un noyau et d'une structure pour les modèles de tâches orientés utilisateurs. *Proceedings of the 17th conference on 17ème Conférence Francophone sur l'Interaction Homme-Machine - IHM 2005*, pp.83–90.

Lê, L.S., Ghose, A. & Morrison, E., 2010. Definition of a description language for business service decomposition. In *Proceedings of First International Conference on Exploring Services Sciences (IESS 1.0), Geneva, Switzerland*. Springer, pp. 96–110.

MacKenzie, C.M. et al., 2006. Reference Model for Service Oriented Architecture 1.0. *Architecture*, 2006(October), pp.1–31.

Maglio, P.P. & Spohrer, J., 2007. Fundamentals of service science. *Journal of the Academy of Marketing Science*, 36(1), pp.18–20.

Malone, T.W. & Crowston, K., 1994. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), pp.87–119.

March, S., 1995. Design and natural science research on information technology. *Decision support systems*.

Marjanovic, O. et al., 2007. Collaborative practice-oriented business processes Creating a new case for business process management and CSCW synergy. *2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2007)*, pp.448–455.

Martin, R.C., 2003. *Agile Software Development: Principles, Patterns, and Practices*, Prentice Hall PTR.

Masserey, G. & Samaan, K., 2006. Implémentation du modèle AMF. In *IHM "06 Proceedings of the 18th International Conferenceof the Association Francophone d"Interaction Homme-Machine*.

Mintzberg, H., 1989. *Le management: voyage au centre des organisations*, Paris: Lavoisier.

Misra, S. & Kumar, V., 2005. Goal-oriented or scenario-based requirements engineering technique - what should a practitioner select? *Canadian Conference on Electrical and Computer Engineering, 2005.*, (May), pp.2288–2292.

Mori, G., Paterno, F. & Santoro, C., 2002. CTTE: support for developing and analyzing task models for interactive system design. *IEEE Transactions on Software Engineering*, 28(8), pp.797–813.

Nitithamyong, P. & Skibniewski, M.-J., 2007. Key success/failure factors and their impacts on system performance of web-based project management systems in construction. *ITcon*, pp.39–59.

Noran, O., 2000. Business modelling: UML vs. IDEF. *School of CIT, Griffith University*. Available at: http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Business+Modelling+:+UML+vs.+IDEF#0 [Accessed June 20, 2012].

Norman, D., 1986. User-centred System Design: New Directions in Human-Computer Interaction.

Nuseibeh, B. & Easterbrook, S., 2000. Requirements engineering: a roadmap. In *Proceedings of the Conference on the Future of Software Engineering*. ACM, pp. 35–46.

Oaks, P., Ter Hofstede, A.H.M. & Edmond, D., 2003. Capabilities: Describing what services can do. In *Service-Oriented Computing-ICSOC 2003*. Springer.

Olsen, G., 2004. *Personas creation and usage toolkit*.

Olson, J., 1996. Groupware in the wild: lessons learned from a year of virtual collocation. In *Proceedings of the 1996 ACM conference on*. pp. 419–427.

Osterwalder, A., 2004. *The business model ontology: A proposition in a design science approach*. Ecole des HEC de l'Université de Lausanne.

O'Farrell, P., 1991. An interaction model of business service production and consumption. *British Journal of Management*.

O'Regan, G., 2008. Computer Programming Languages. In *A Brief History of Computing*.

O'Sullivan, J.J., 2006. *Towards a precise understanding of service properties*. Queensland University of Technology.

Paetsch, F., Eberlein, a. & Maurer, F., 2003. Requirements engineering and agile software development. *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pp.308–313.

Papazoglou, M.P. & Georgakopoulos, D., 2003. Service-oriented computing. *Communications of the ACM*, 46(10), pp.25–28.

Papazoglou, Michael P. & Heuvel, W.-J. Van Den, 2006. Service-oriented design and development methodology. *International Journal of Web Engineering and Technology*, 2(4), p.412.

Paterno, F., 2001. Towards a UML for interactive systems. In M. Reed Little & L Nigay, eds. *Engineering for human-computer interaction*. Springer-Verlag Berlin Heidelberg 2001, pp. 7–18.

Paternò, F., Mancini, C. & Meniconi, S., 1997. ConcurTaskTrees: A diagrammatic notation for specifying task models. In *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction*. Citeseer, pp. 362–369.

Peffers, K. et al., 2007. A Design Science Research Methodology for Information Systems Research. *Journal of Management Information Systems*, 24(3), pp.45–77.

Peixoto, D., Batista, V. & Atayde, A., 2008. A comparison of bpmn and uml 2.0 activity diagrams. In *VII Simposio Brasileiro de Qualidade de Software*.

Pereira, CM, 2004. A method to define an Enterprise Architecture using the Zachman Framework. In *Proceedings of the 2004 ACM symposium on Applied computing*. ACM.

Pinelle, D. & Gutwin, C., 2005. A groupware design framework for loosely coupled workgroups. In *ECSCW 2005*. pp. 65–82.

Piquet, A., 2009. *Guide pratique du travail collaboratif: Théories , méthodes et outils au service de la collaboration*,

Pribeanu, C., 2005. An Approach to Task Modeling for User Interface Design. *World Academy of Science, Engineering and Technology*, 5, pp.5–8.

Pruitt, J. & Grudin, J, 2003. Personas: practice and theory. In *Proceedings of the 2003 conference on Designing for user experiences*. ACM.

Pérochon, L., 2008. *Ingénierie dirigée par les modèles Model Driven Architecture*,

Quartel, D., Dijkman, R. & Van Sinderen, M., 2004. Methodological support for service-oriented design with ISDL. *Proceedings of the 2nd international conference on Service oriented computing - ICSOC '04*

Rameau, G. & Samyn, E., 2006. LE TRAVAIL COLLABORATIF ASSISTE PAR ORDINATEUR (TCAO) Exemple d'une solution technologique avec X-TEK. In *23 ème congrés de l'AIPU*. Monastir, pp. 1–22.

Ramollari, E. & Dranidis, D., 2007. A survey of service oriented development methodologies. In *Workshop on Service Oriented*. pp. 75–80.

Rosemann, M. et al., 2009. Business Service Management. *Service Management*, p.16.

Saffin, S. & Leclercq, P., 2010. Studio digital collaboratif, un environnement multimodal de conception collaborative a distance: application et perspectives. In *IHM'10 - Atelier Collecticiels*. pp. 16–18.

Safin, S., Kubicki, S. & Hanser, D., 2012. Enseigner la co-conception à distance: Retour sur cinq années d'expérience. In *SCAN12 - Complexité (s) des modèles de l'architecture numérique*. Presses Universitaires de Nancy.

Salber, D. et al., 1995. De l'observabilité et de l'honnèteté: le cas du controle d'accès dans la Communication Homme-Homme Médiatisée. In *IHM'95*.

Salinesi, C., 2004. Authoring Use Cases. *Scenarios and Use Cases, Stories through the System Life-Cycle*.

Salvador, T., Scholtz, J. & Larson, J., 1996. The Denver model for groupware design. *ACM SIGCHI Bulletin*, 28(1), pp.52–58.

Samaan, K., 2006. *Prise en Compte du Modèle d'Interaction dans le Processus de Construction et d'Adaptation d'Applications Interactives*. Ecole dentrale de Lyon.

Sandkuhl, K., 2010. Supporting Collaborative Engineering with Information Supply Patterns. In *18th Euromicro Conference on Parallel, Distributed and Network-based Processing*. Ieee, pp. 375–384.

Santos, L., Lopes, J. & Leitao, A., 2012. Collaborative Digital Design - When the architect meets the software engineer. In H. Achten et al., eds. *30th eCAADe conference vol.2 physical digitality*. Prague, pp. 87–97.

Satyanarayanan, M., 1996. Fundamental challenges in mobile computing. In *Proceedings of the fifteenth annual ACM symposium on Principles of distributed computing - PODC '96*. New York, New York, USA: ACM Press.

Saïkali, K., 2001. *Flexibilité des Workflows par l'approche objet: 2FLOW, un framework pour Workflows flexibles*. Ecole centrale de Lyon.

Schmidt, K. & Simone, C., 1996. Coordination mechanisms: Towards a conceptual foundation of CSCW systems design. *Computer Supported Cooperative Work (CSCW)*, 5(2-3), pp.155–200.

Schmidt, K. & Wagner, I., 2004. Ordering Systems: Coordinative Practices and Artifacts in Architectural Design and Planning. *Computer Supported Cooperative Work (CSCW)*, 13(5-6), pp.349–408.

Schwaber, K & Sutherland, J., 2011. *The scrum guide*

Schwaber, Ken & Beedle, M., 2001. *Agile Software Development with Scrum*, Prentice Hall.

Seffah, A., Kolski, C. & Idoughi, D., 2009. Persona comme outil de design de services interactifs: principes et exemple en e-maintenance. In *IHM'09*. Grenoble, France: ACM, pp. 13–16.

Sierhuis, M. & J.Clancey, W., 2002. Modeling and Simulating Work Practice: A Method for Work Systems Design. *IEEE Intelligent Systems*, 17(05), pp.32–41.

Simon, H.A., 2004. *The sciences of artificial*, MIT Press.

Sohlenkamp, M., Prinz, W. & Fuchs, L., 2000. POLIAwaC: Design and evaluation of an awareness-enhanced groupware client. *AI & SOCIETY*, 14, pp.31–47.

Sommerville, I., 2005. Integrated requirements engineering: A tutorial. *Software, IEEE*, 22(1), pp.16–23.

Sommerville, I., 1996. *Introduction to Software Engineering*

Sottet, J.-S., Calvary, G. & Favre, J.-M., 2005. Ingénierie de l'Interaction Homme-Machine Dirigée par les Modèles. *IDM, Premières Journées sur l'Ingénierie Dirigée par les Modèles*.

Soulier, E. & Lewkowicz, M., 2006. Simulation des pratiques collaboratives pour la conception des SI basés sur les processus métier. *Revue des Sciences et Technologies de ll'Information*, 11(3), pp.73–94.

Soulier, E., Lewkowicz, M. & Corouge, N., 2007. Gestion des processus métier et travail collaboratif. *myriam.lewkowicz.free.fr*, p.25.

Suchman, L., 1987. *Plans and Situated Actions: The Problem of Human-Machine Communication (Learning in Doing: Social, Cognitive and Computational Perspectives)*, Cambridge University Press.

Sutcliffe, A., 2005. Applying small group theory to analysis and design of CSCW systems. *ACM SIGSOFT Software Engineering Notes*, pp.1–6.

Sutcliffe, A., 2007. Requirements Engineering from an HCI Perspective. In *Requirements Engineering Activities and Processes*. pp. 1–39.

Sutcliffe, A., 2003. Scenario-based requirements engineering. *Journal of Lightwave Technology*, pp.320–329.

Tarpin Bernard, F., 1997. *Travail coopératif synchrone assisté par ordinateur : Approche AMF-C*. Ecole centrale de Lyon.

Tazari, M., Grimm, M. & Finke, M., 2003. Modelling user context. *Proceedings of 10th International Conference on Human-Computer Interaction*.

Tellioglu, H., 2006. Coordination Design. In *Proceedings of the 20th International Conference on Advanced Information Networking and Applications*. 1: IEEE Computer Society, pp. 425–432.

Thevenin, D, 2001. *Adaptation en Interaction Homme-Machine : le cas de la Plasticité*.

Touzi, J. et al., 2009. A model-driven approach for collaborative service-oriented architecture design. *International Journal of Production Economics*, 121(1), pp.5–20.

Vaishnavi, V. & Kuechler, W., 2007. *Design science research methods and patterns: innovating information and communication technology*, Auerbach Publications.

Vanderdonckt, JM & Bodart, F, 1993. Encapsulating knowledge for intelligent automatic interaction objects selection. *Proceedings of INTERACT'93*

Veer, G. Van Der et al., 2000. Task based groupware design: putting theory into practice. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*. New York, NY, USA: ACM, pp. 326–337.

Veer, G. van der, 1996. *The Design of Computer Supported Cooperative Work and Groupware Systems*, North Holland Elsevier Science.

Vissers, C., Lankhorst, M. & Slagter, R., 2004. Reference models for advanced e-services. In *Digital Communities in A Networked Society*. Springer, pp. 369–393.

White, S., 2004. *Introduction to BPMN*,

Wieringa, R., 1998. A survey of structured and object-oriented software specification methods and techniques. *ACM Computing Surveys (CSUR)*, 30(4), p.69.

Williams, A., 2009. User-centered design, activity-centered design, and goal-directed design: a review of three methods for designing web applications. In *SIGDOC'09*. Bloomington, Indiana: ACM, pp. 1–8.

Winckler, M., Palanque, P. & CMDS, 2004. Tasks and scenario-based evaluation of information visualization techniques. *Proceedings of the 3rd conference on Task models and diagrams* pp. 165-172.

Yamin, R. & Harmelink, D., 2001. Comparison of linear scheduling model (LSM) and critical path method (CPM). *Journal of Construction Engineering And Management*, 127(5), pp.374–381.

Yeh, R.T. & Zave, P., 1980. Specifying software requirements. *Proceedings of the IEEE*, 68(9), pp.1077–1085.

Yin, C., 2010. *Samcco: un Système d'Apprentissage Mobile Contextuel et Collaboratif dans des Situations Professionnelles*. Ecole Centrale de Lyon.

Zachman, J. a., 1987. A framework for information systems architecture. *IBM Systems Journal*, 26(3), pp.276–292.

Zhang, J. et al., 2012. Development and Implementation of an Industry Foundation Classes-Based Graphic Information Model for Virtual Construction. *Computer-Aided Civil and Infrastructure Engineering*.

Zhang, Z., Liu, R. & Yang, H., 2005. Service identification and packaging in service oriented reengineering. In *Proceedings of the 17th International Conference on Software Engineering and Knowledge Management*. pp. 219–226.

Zignale, D., Kubicki, S., Ramel, S., et al., 2011. A model-based method for the design of services in collaborative business environments. In *Proceedings of IESS 1.1: Second International Conference on Exploring Services Sciences*. Geneva, Switzerland, p. 15.

Zignale, D., Kubicki, S. & Halin, G., 2011. Business practices analysis for the adaptation of IT services to AEC projects. Case study of design assessment related practices. In *CIB w078-w102 2011*. Sophia-Antipolis, France.

Zimmermann, O., 2004. Elements of service-oriented analysis and design. *IBM developerworks*.

# Glossaire

## Activité :

L’activité (humaine) est un ensemble de tâches dirigées vers une finalité et organisées au sein d'un processus, notamment dans le cadre de l’exercice d’une fonction, d’un métier. La théorie de l’activité la décompose en trois sous-ensembles : l’activité, l’action et l’opération. Le projet AIC est une activité : c’est un double processus composé d’une étape préparatoire (liée à la conception) et d’une étape opératoire (liée à la construction).

## Artefact :

L’artefact est un produit artificiel réalisé par l’homme. Dans cette étude, il regroupe les documents relatifs au domaine AIC (les plans, les descriptifs textes, les photos…) ainsi que les objets (c.à.d. les ouvrages ou éléments d’ouvrages architecturaux).

## Collecticiel :

Le collecticiel est un logiciel dédié au support de l’activité collective que ce soit par l’assistance à la communication, la coordination ou la co-production, de manière synchrone ou asynchrone, co-localisée ou à distance.

## Contexte :

Le contexte est un ensemble de circonstances déterminées par un environnement ou une situation, et déterminantes dans le comportement de quelque chose ou quelqu’un.

## Méthode de conception :

Une méthode de conception est un processus créatif itératif composé de plusieurs étapes : la connaissance du besoin, la suggestion, la proposition, la validation et la conclusion. Elle est supportée par modèles et des outils et associe plusieurs points de vue.

**Modèle :**

Un modèle est la représentation de quelque chose à travers, sous forme d'objet physique (ex. une maquette) ou de description (un texte, un dessin…). Un modèle peut servir à la description (ex. le plan d'une ville) comme à la conception (ex. le plan d'une maison à construire). Un modèle abstrait ce qu'il représente à travers des concepts.

**Outil :**

Physique ou numérique, l'outil assiste son usager dans l'exécution d'une ou plusieurs tâches. L'adaptation d'un outil se mesure par son utilité (sa capacité à répondre au besoin de l'usager) et son utilisabilité (relative à la facilité, la rapidité d'utilisation ou encore le besoin d'apprentissage réduit).

**Paradigme / Point de vue :**

Le paradigme ou point de vue est la manière de voir ou de considérer quelque chose. Il est relatif à l'interprétation d'un contenu et au rapport entre l'acteur (qui a ce point de vue) et l'objet de son étude.

**Pratique (collective et individuelle) :**

La pratique est l'exercice d'un métier, une manière de travailler, un comportement habituel avec une finalité. Les pratiques dites collectives (PC), à savoir relatives à un objectif commun à plusieurs personnes, sont décomposées en pratiques individuelles (PI) exécutées par chacune de ces personnes. L'exécution de toutes les PI qui composent une PC doit permettre d'atteindre l'objectif de cette dernière.

**Service (et service informatique) :**

Un service est une prestation qui met à disposition d'un client une capacité technique ou intellectuelle pour supporter un besoin. En informatique, un service est une fonctionnalité ou partie de fonctionnalité mise à disposition par un composant logiciel pour assurer une tâche particulière.

**Usage :**

Le terme usage définit une utilisation commune, normale, prévue alors que l'utilisation se rapporte plutôt à un point particulier, une situation donnée. Par extension, l'usage se réfère à l'appropriation. Pour caractériser la médiatisation d'une pratique métier par l'emploi d'un outil, nous utilisons le terme d'usage plutôt qu'utilisation car il couvre une définition plus large que celle du simple emploi. On cherche à retrouver, en plus de la notion d'emploi, les notions d'habitude, d'objectif et d'appropriation par les acteurs.

.

# Index

# Table des illustrations

## Liste des figures

## Liste des tableaux

# Annexes

## Analyse des « tickets » de conception de CRTI-weB

L'analyse évoquée au § 1.5.3 avait pour but d'évaluer le nombre de développements sur les services CRTI-weB en rapport avec un « besoin métier », à savoir une exigence issue du domaine professionnelle et non liée à des problèmes d'utilisabilité.

D'après l'analyse menée, la moitié des développements initiés ont pour origine un besoin métier clairement identifié. Nous avons également pu évaluer les modifications impliquées dans l'usage (générale, sur les tâches, sur le contexte ou sur le contenu)

| n° du Ticket | Nom/description de la demande | Y a-t-il une pratique métier identifiée? | | Quelle modification dans l'usage? | | Quelle amélioration dans l'utilisabilité? Quel apport de la fonctionnalité? |
|---|---|---|---|---|---|---|
| | | Oui/non | Laquelle? | G/T/C/Cu | Détail | |
| 12 | Chargement des fichiers "en arrière plan" | Oui | Partager des groupes de documents différents | Générale | Usage plus rapide | Plus de confort dans l'upload, gain de temps |
| 49 | Désactiver onglet d'archivage | Non | | Générale | Usage supprimé | Pas d''onglet inutile superflu |
| 50 | Journalisation des courriels de notification | Oui | Etre averti des opérations de ses | Générale | Modification de la régularité | Mails moins dispersés |
| 51 | Ajouter date de dépôt de l'indice | Oui | Retrouver un document | Contenu | Identification de la | Meilleure compréhension, gain de |
| 52 | Résoudre le Pb de gestion des noms de fichiers existants quand on change la CN au | Non | | Contenu | Changement du nom | |
| 53 | Améliorer l'onglet Historique avant de le rendre à nouveau disponible | Non | | | | Meilleure ergonomie |
| 63 | Nouvelle mise en forme du pdf du compte-rendu selon demande de l'OAI. | Oui | Créer et partager des CR de chantier | Contenu | Modification du format de la donnée | Meilleure compréhension, gain de temps |
| 115 | pouvoir identifier les acteurs qui ont accédé à un compte-rendu particulier | Oui | Identifier activité des collaborateurs | Tâche | Identifiation des acteur ayant consulté un | Meilleure compréhension, gain de temps |
| 116 | Comptes-rendus : classer les acteurs | Oui | Identifier activité des | Tâche | Classement des acteurs | Meilleure compréhension, gain de |
| 117 | CR : classer les remarques par priorité dans | Oui | Créer un CR de chantier | Tâche | Classement des | Meilleure compréhension, gain de |
| 118 | CR : Pouvoir identifier rapidement les nouvelles remarques | Oui | Identifier activité des collaborateurs | Tâche | Identifier remarques depuis dernière | Meilleure compréhension, gain de temps |
| 119a | Lors de l'ajout d'un niveau d'historique (MaJ d'une remarque), il est nécessaire de | oui | Identifier les commentaires des | Tâche + contenu | Identifier date de remarque | Meilleure compréhension, gain de temps |
| 119b | Lors de l'ajout d'un niveau d'historique, il est nécessaire de pouvoir modifier les | | | | | |
| 119c | Lorsqu'on effectue un enregistrement et que l'on revient dans la liste des remarques, il faut repositionner la fenêtre | Non | | NIL | | Gain de temps, plus de confort |
| 119d | la modification d'une remarque (ajout d'historique) devrait pouvoir se faire directement dans la liste des remarques | Non | | Tâche | Tâche d'ouverture du détail de la remarque supprimée | Gain de temps, plus de confort |
| 120 | Les photos devraient pouvoir être intégrées à la version PDF du compte- | Oui | Partager CR de chantier avec des photos | Tâche + Contenu | Intégrer photo | Plus d'informations partageables, Centralisation des informations |
| 121 | L'ergonomie de la visionneuse de photos | Non | | NIL | | Meilleure ergonomie |
| 123 | Archivage des données en fin de projet | Oui | Récupérer toutes les | Générale | Nouvel usage archivage | Nouvelle fonctionnalité utile |
| 124 | Adaptation des Web Services | Non | | NIL | | Amélioration des fonctionnalités |
| 131 | Modification des templates - Ajout d'id pour tests automatiques Selenium | Non | | NIL | | Destiné aux développeurs |
| 149 | Externaliser le code JS d'un fichier .tpl | Non | | NIL | | Destiné aux développeurs |
| 150 | Créer un web service pour l'upload d'un ou plusieurs documents pour un projet | Non | | NIL | | Destiné aux développeurs |
| 160 | Externaisation méthode vérification CN au chargement d'un document | Non | | NIL | | Destiné aux développeurs |
| 161 | Pour sauvegarder la date d'ajout d'un historique de remarque, il manque un | Non | | NIL | | Destiné aux développeurs |
| 167 | Lorsqu'on veut mettre une action sur un document, il faut limiter les organismes | Oui | Avertir certains collaborateurs d'un | Contenu | Les acteurs sélectionnés sont | Pas de notifications inutiles |
| 173 | Ajout du WebService permettant de checker le nom fonction du chantier et de | Non | | NIL | | Destiné aux développeurs |
| 180 | Lors de l'ajout d'un utilisateur ou de la modification, rendre obligatoire la | Oui | Impliquer un acteur dans le projet | Contenu | Assignation du rôle obligatoire | Pouvoir reconnaitre les utilisateurs |
| 185 | Le but est de supprimer une action via le | Non | | NIL | | Destiné aux développeurs |

*Figure 143. Grille d'analyse des tickets (partie 1/ 4)*

| | | | | | | |
|---|---|---|---|---|---|---|
| 199 | Retour en fin de page après avoir ajouté un | Non | | NIL | | Gain de temps, plus de confort |
| 201 | Dans les vues affichant les remarques, il manque une icône pour remplacer le libellé "Clôturer" qui indique qu'une | Non | | NIL | | Meilleure lecture de l'information |
| 202 | Affectation rôle à un utilisateur lors de | Oui | Impliquer un acteur dans | Contenu | Assignation du rôle | Pouvoir reconnaitre les utilisateurs |
| 204 | Envoie notification lors suppression document et changement de zone | Oui | Avertir certains collaborateurs d'une | Tâche | le système envoie une notification | Infomration mieux transmise |
| 209 | CN affichée est celle de l'OAI et pas celle | Non | | NIL | | Rectification d'une erreur |
| 211 | Mise à jour page de garde (accès plaquette et mise en valeur du contact) | Non | | Générale | Nouvel usage consultation des infos | Accès à une information utile + Meilleure lisibilité |
| 227 | Interdire la création d'un compte-rendu si pas de réunion liée | Oui | Rédiger un CR après le déroulement d'une | Générale | Usage succèdant obligatoirement à un | Pas d'incohérence dans l'information |
| 231 | Ajout support GIF dans clearbircks- | | | Contenu | Format de fichier | Plus de possibilités de partage |
| 237 | statistique : mesurer le temps de réaction à un document suite à sa mise en ligne | Non | | NIL | | Destiné aux développeurs |
| 238 | statistiques sur temps d'accès aux documents et actions suite au dépot de | Non | | NIL | | Destiné aux développeurs |
| 240 | Statistiques : intégration des informations de la classe stats.class.php | Non | | NIL | | Destiné aux développeurs |
| 241 | Traductions | Non | | Contexte | Utilisateur avec un language différent | Accessibilité aux utilisateurs d'autre nationalité |
| 246 | export d'un chantier complet de CRTI-weB hors gestion CR, sous format XML pour futur réimport et format HTML pour les | Oui | Récupérer toutes les données en fin de projet | Contenu | Définition du format de données | Nouvelle fonctionnalité utile |
| 257 | Version en Allemand de CRTI-weB | Non | | Contexte | Utilisateur avec un language différent | Accessibilité aux utilisateurs d'autre nationalité |
| 258 | Dans un compte-rendu, les sections "Documents à jour" et "Echange de | Non | | NIL | | Rectification de fonctionnalités non opérationnelles |
| 259 | Rajouter dans l'export tous les fichiers nécessaire pour le client "reader" (index.html, SRC, ...) de façon à ce que l'on | Non | | NIL | | Destiné aux développeurs |
| 260 | Export Chantiers - Remplacer la copie des fichiers exportés par des liens | | | | | |
| 270 | Favicon : petit icône pour affichage en | Non | | NIL | | Détail graphique |
| 276 | Export chantier : prendre en compte les | Oui | Récupérer toutes les | Contenu | Définition du contenu | Nouvelle fonctionnalité utile |
| 277 | Export CR : modification méthode genererPDF pour permettre sauvegarde | Non | | NIL | | Destiné aux développeurs |
| 278 | Intégrer l'export des CR dans le "Archives | Non | | NIL | | Destiné aux développeurs |
| 280 | Le #120 concernait l'étude de faisabilité. Ce ticket distinct correspond à la | Oui | Partager CR de chantier avec des photos | Tâche + Contenu | Intégrer photo | Plus d'informations partageables, Centralisation des informations |
| 283 | Actions sur document : filtrage personnes du groupe sur base de la zone du | Oui | Identifier activité des collaborateurs | Tâche + Contenu | Identifier acteurs ayant ajouté une action sur un | Meilleure compréhension, gain de temps |
| 284 | Ajout/mise à jour de document, version multi-uploads : ajout groupe pour les | Oui | Avertir certains collaborateurs d'un | Contenu | Selection par groupe | Gain de temps |
| 285 | si une personne essaye d'ouvrir un fichier auquel ne peut normalement pas accéder, afficher un message indiquant que ce document n'est pas ou plus accessible, et | Non | | Tâche | Tâche système d'avertissement | Meilleure compréhension |
| 287 | faire apparaître une liste des organismes supprimés pour réactivation. | | | | | |
| 293 | Suppression organisme : ne pas afficher la croix si organisme non vide | Non | | NIL | | Suppression d'une icône inutile |
| 296 | Permettre importation utilisateur préalablement complètement supprimé | Non | | Tâche + Contenu | Traiter acteurs supprimé | Gain de temps |
| 299 | Importation organismes préalablement complètement supprimé | Non | | Tâche + Contenu | Traiter acteurs supprimé | Gain de temps |
| 307 | Recherche de remarques : lien vers les remarques trouvées | Oui | Consulter les commentaires des | Tâche | Ouvrir document après recherche | Gain de temps |

*Figure 144. Grille d'analyse des tickets (partie 2/ 4)*

| | | | | | | |
|---|---|---|---|---|---|---|
| 320 | Annuaire des participants au projet au format PDF | Oui | Consulter la liste des collaborateurs | Général | Nouvel usage d'export au format pdf | Information dans un format approprié à l'échange |
| 322 | Dépôt document : possibilité dépôt au niveau supérieur | Oui | Partager des informations avec un | | | |
| 326 | Création/import organisme : associer la CN "OAI" par défaut | Oui | Impliquer un acteur ou un groupe dans le projet | Tâche + Contenu | Convention de nommage prédéfinie | Gain de temps |
| 327 | Service CR : modération commentaires | Non | | Général | Pouvoir gérer les commentaires | management facilité |
| 328 | Service Document : modération réactions | Non | | Général | Pouvoir gérer les commentaires | management facilité |
| 329 | CN : ajout par chargement contenu fichier | Non | | Tâche + Contenu | Convention de nommage importée par | |
| 335 | Proposition de l'upload multiple optionnelle | Oui | Partage de documents tantôt individuellement | Contenu | Choix entre Fichier ou groupe de fichier | Choix accordé entre 2 fonctionnalités |
| 345 | Ajout librairie PHPExcel nécessaire pour importation métadonnées en lot depuis | Non | | NIL | | Destiné aux développeurs |
| 351 | A la création d'une remarque, le cadre de description est beaucoup trop petit | Non | | NIL | | Détail graphique |
| 352 | Lorsqu'une personne est ajoutée pour la diffusion du compte-rendu, elle est obligatoirement « Présente, Absente ou Excusée », nous voudrions qu'il soit | Oui | Partager un CR avec les collaborateurs de son choix | Contenu | Limiter l'information apportée à la diffusion | Suppression d'une fonctionnalité inutile |
| 353 | Suppression priorité par défaut d'une remarque | Non | | Contenu | Limiter l'information apportée à une | Suppression d'une fonctionnalité inutile |
| 369 | Cacher l'administrateur de la listes des personnes présentent dans l'annuaire des | Non | | NIL | | Suppression d'une information |
| 387 | Permettre de trier les remarques par date de création ou par organisme | Oui | Retrouver les remarques faites sur un CR | Tâche + contenu | Nouvel usage de recherche des | Meilleure compréhension, gain de temps |
| 389 | Le filtrage des remarques associées à plusieurs organismes ne devrait faire | | | | | |
| 391 | Modifier ergonomie de l'ajout de PDFs | Non | | NIL | | Meilleure ergonomie |
| 398 | récupérer les notes introduites dans le précédent CR. | Non | | Contenu | Partie du document récupérée par le | Gain de temps |
| 403 | Il manque les icônes d'action pour Compte | Non | | NIL | | Détail graphique |
| 435 | Ajouter la date de la mise à jour de l'indice pour plus de visibilité | Oui | Mettre à jour un document de travail | Tâche + Contenu | Identifier les dates de MaJ | Meilleure compréhension, gain de temps |
| 436 | Modifier l'image par défaut des projets ne | Non | | NIL | | Détail graphique |
| 440 | Intégration avec AutoCAD WS | Oui | Consulter les documents partagés | Contenu + Contexte | Utilisation de l'explorateur web pour | Adaptation à l'utilisation d'un terminal mobile |
| 444 | PROMOBE: Créer des templates de convention de nommage | Non | | Tâche + Contenu | Sauvegader des CN types | Gain de temps |
| 445 | PROMOBE: Création du rôle Administrateur projet | Oui | Plusieurs pratiques de gestion menées par un | Contexte | Nouveau role opérationnel | nouveau type d'utilisateur |
| 447 | PROMOBE: Ajout de nouveaux organismes disponible uniquement dans la gestion d'un projet, pas possible dans la partie | Non | | Tâche + Contenu | Ajout d'organismes limité | Modification d'une fonctionnalité |
| 448 | PROMOBE: Possibilité d'éditer les informations des organismes créés | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Edition d'organisme limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 449 | PROMOBE: Suppression des organismes créés uniquement par l'administrateur | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Suppression d'organisme limité à un | Attribution d'une fonctionnalité à un type d'utilisateur |
| 450 | PROMOBE: L'administrateur projet doit pouvoir ajouter des utilisateurs, et | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Ajout et import d'utilisateurs limité à | Attribution d'une fonctionnalité à un type d'utilisateur |
| 451 | PROMOBE: Suppression des utilisateurs créés uniquement par l'administrateur | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Suppression d'utilisateurs limité à | Attribution d'une fonctionnalité à un type d'utilisateur |
| 452 | PROMOBE: Possibilité d'éditer les informations des utilisateurs créés | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Edition d'utilisateurs limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 453 | PROMOBE: L'administrateur d'un projet doit pouvoir modifier les zones d'un projet | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Modifications des zones limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 454 | PROMOBE: L'administrateur projet doit pouvoir renvoyer un mot de passe | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Renvoi de mot de passe limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |

*Figure 145. Grille d'analyse des tickets (partie 3/ 4)*

| | | | | | | |
|---|---|---|---|---|---|---|
| 458 | PROMOBE: Possibilité pour l'administrateur projet de voir tous les | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Accès à tous les documents limité à un | Attribution d'une fonctionnalité à un type d'utilisateur |
| 459 | PROMOBE: Possibilité pour l'administrateur projet de voir tous les CR | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Accès à tous les CR limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 463 | Prise en compte du 'ç' dans les adresses | Non | | NIL | | Détail fonctionnel |
| 464 | Actuellement, l'administrateur CRTI-weB peut ajouter autant de groupes qu'il veut, mais ne peut pas les éditer où les supprimer. Il faut développer | Oui | Plusieurs pratiques de gestion menées par un seul acteur. | Contexte + Tâche | Edition et modifications d'e groupes par un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 465 | Création du répertoire d'upload d'un projet | Non | | NIL | | Destiné aux développeurs |
| 466 | Possibilité d'annuler l'abonnement sur un groupe de documents | Oui | Vérifier ou pas l'évolution d'un | Tâche | Possibilité d'annuler une tâche | |
| 471 | Export PDF d'une liste des documents | Oui | Consulter la liste des documents partagés | Général | Nouvel usage d'export au format pdf | Information dans un format approprié à l'échange |
| 478 | Envoyer un mail à la suppression d'un | Oui | Avertir les | Tâche | Tâche système de | Meilleure compréhension, gain de |
| 483 | Ajouter la possibilité d'attacher un document dans le detail d'un projet | Oui | Partager des documents du projet | Tâche + contenu | Ajouter un document accessible depuis le | Transmission d'informations supplémentaires |
| 493 | Empêcher la modification d'un organisme par un utilisateur simple | Oui | Plusieurs pratiques de gestion menées par un | Contexte + Tâche | Edition d'organisme limité à un utilisateur | Attribution d'une fonctionnalité à un type d'utilisateur |
| 495 | Création d'un SMTP local pour forwarder les non-delivery failure à l'envoi de mail | Non | | NIL | | Destiné aux développeurs |
| 496 | Création d'une partie "Statistiques CRTI- | Non | | NIL | | Destiné aux développeurs |
| 510 | Génération d'un fichier Excel permettant de faciliter la facturation en fin de mois (de | Oui | Partager des informations nécessaires | Général | Nouvel usage d'export au format excel | Nouvelle fonctionnalité utile |
| 511 | Rajouter une option "Type de facturation" | Oui | Partager des informations nécessaires | Contenu | Choix d'un type de facturation | Gain de temps, fonctionnalité utile |
| 520 | Afficher les projets auquels appartiennent une personne | Oui | Consulter les activités d'un collaborateur | Tâche + Contenu | Identifier les projets d'un acteur | Meilleure compréhension, gain de temps |
| 521 | Afficher les personnes qui sont contenues dans un organisme | Oui | Consulter les activités d'un collaborateur | Tâche + Contenu | Identifier les personnes d'un organisme | Meilleure compréhension, gain de temps |
| 525 | Il faudrait ajouter la possibilité pour l'administrateur (voir l'administrateur | Non | | Tâche + Contenu | modifier auteur d'un document | Modification d'information |
| 533 | Pour répondre aux questions récurrentes des utilisateurs, il serait bien de créer une | Non | | Général | Nouvel usage de consultation d'une FAQ | Informations utiles |
| 534 | Renommer les zones permettant de gérer la visibilité des documents dans CRTI-weB: | | | | | |
| 535 | Installation logiciel wiki | Non | | Général | Nouvel usage de | Informations utiles |
| 536 | Rédiger les questions à mettre dans la FAQ | Non | | Général | Nouvel usage de | Informations utiles |
| 538 | Il faudrait mettre l'adresse mail de l'administrateur projet et non plus "info.crti-web@kitry.lu" dans le "From" | Non | | NIL | | Détail fonctionnel |
| 539 | Ajouter Icône "Modifier le propriétaire" | Non | | NIL | | Détail graphique |
| 540 | L'icône "Ajouter un planning" global ou semaine n'est pas joli, il faudrait en créer | Non | | NIL | | Détail graphique |
| 544 | Sauvegarder un filtrage de documents | Oui | Retrouver les documents d'un projet | Tâche | Tâches de sauvegarde et récupération | Fonctionnalité utile |
| 545 | Il faut envoyer un mail à l'utilisateur et au chef de projet lorsqu'une personne est | Oui | Avertir les collaborateurs des | Tâche | Tâche système de notification | Information mieux transmise |
| 546 | PROMOBE : Lors de la création d'un utilisateur, envoyer une copie de l'email à | Oui | Avertir les collaborateurs des | Tâche | Tâche système de notification | Information mieux transmise |
| 547 | Envoi d'un mail au chef de projet lors du dépôt d'un document | Oui | Avertir les collaborateurs des | Tâche | Tâche système de notification | Information mieux transmise |
| 548 | mettre l'émetteur de notre choix lors de | Non | | NIL | | Détail fonctionnel |
| 554 | Création d'une page d'accueil | Non | | NIL | | Détail graphique |
| 556 | Il faudrait pouvoir sauvegarder les recherches. | Oui | Retrouver les documents d'un projet | Tâche | et récupération d'infromation de filtre | Fonctionnalité utile |

| Pourcentage de "oui" |
|---|
| 48,67% |

*Figure 146. Grille d'analyse des tickets (partie 4/ 4)*

**Cahier d’exigences : mode d’emploi**

Les pages qui suivent constituent le mode d’emploi du cahier d’exigences tel qu’il est actuellement proposé. Il a pour but d’assister les experts métier, concepteurs et développeurs dans leurs phases de modélisation respectives et dans la lecture générale du document.

# Cahier d’exigences pour le développement de services informatiques spécifiques

## Mode d’emploi

1. **Description du document**

   a. **But du document**

Ce document a pour but de définir les exigences qui guideront le développement de services informatiques. Il devra décrire synthétiquement les pratiques métiers du client ainsi que les usages induits qu’il adoptera en tant qu’utilisateur du système à développer (ou améliorer). Ces éléments auront été identifiés par l’analyste métier grâce à l’implication de l’utilisateur lui-même dans la phase d’étude. La combinaison de ces exigences donnera lieu à une spécification technique du service sous forme d’un diagramme UML.

   b. **Portée du document**

Ce cahier des charges servira de base à la conception de services spécifiques. Il est destiné à guider le développeur dans son travail de conception sans qu’il n’ait besoin d’avoir un recours ultérieur aux futurs utilisateurs. Des premiers maquettages serviront de support à la discussion avant de passer au prototypage de la solution.

   c. **Définitions, Acronymes et Abréviations**

PC (Pratique Collective) : comportement d’un groupe adopté dans le but de répondre à un besoin métier récurrent.
PI (Pratique Individuelle) : comportement d’un individu, impliqué dans une PC, adopté dans le but d’apporter sa contribution à la réponse au besoin métier.
Opération : décomposition d’une PI visant à décrire ce que fait réellement l’acteur.
Artefact : élément produit, utilisé ou référé lors d’une opération, que ce soit un document, un message, un objet physique, une tâche ou tout autre élément support du travail collaboratif.
Usage : médiatisation des pratiques à travers l’utilisation d’une technologie.
Tache : décomposition d’un usage visant à décrire ce que font réellement l’utilisateur et le système.

2. **Organisation du document**

Afin de définir les exigences du service à développer, le document modélisera, à travers des formalismes différents (diagrammes, tableaux, textes, maquettages…) :

Partie 1 : Une analyse métier, décrivant les pratiques collectives et individuelles à outiller

Partie 2 : Une analyse fonctionnelle, décrivant les usages relatifs aux pratiques

Partie 3 : La spécification technique du service

# Partie 1 : pratiques métiers

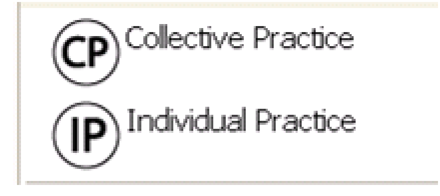
Le diagramme définit / à définir ici schématise une Pratique Collective (Collective Practice, CP) décomposée en une ou plusieurs Pratiques Individuelles (Individual Practices, IP) adoptées par le client ou dans un cas similaire. Identifier les Pratiques Individuelles d'un acteur d'un projet de construction permettra de connaître ses besoins.

***Légende et précisions :***

-

La pratique collective (Collective Practice – CP) est définie par :

- Un nom (name) décrivant en quelques mots en quoi elle consiste
- Une famille (family) choisie parmi un ensemble définit (11 familles possibles)
- Une description ayant pour but d'expliquer plus précisément la nature de la pratique.




La pratique individuelle (Individual Practice – IP) est définie par :

- Un nom (name) décrivant en quelques mots en quoi elle consiste
- Une description ayant pour but d'expliquer plus précisément la nature de la pratique.



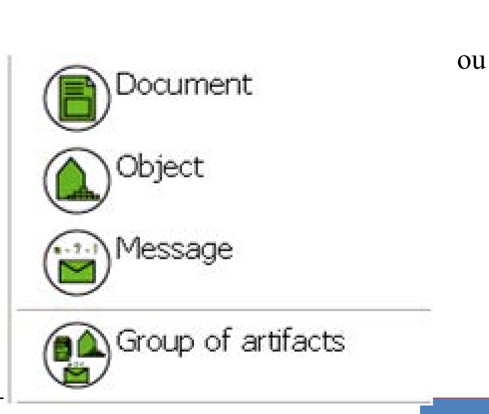

Les opérations(Operations) qui composent une pratique individuelle sont de type :

- « Communication » : Contacter (Contact), Avertir (Advertise), Demander (Ask for), Valider (Validate), Commenter (Comment).
- « Production » : Créer (Create), Modifier (Modify), Mettre à jour (Update), Effacer (Delete)), Exécuter (Execute), Lier (Link), Inclure(Include)
- « Diffusion » (Availability) : Partager (Share), Annuler partage (Unshare)
- « Recherche » (Research) : Rechercher (Look for), Obtenir (Get), Consulter (Consult), Identifier (Identify), Vérifier (Verify).

-

Les artefacts (artifacts), éléments produits, utilisés ou référés lors d'une opération, sont de type :

- Document : géométral (geometral), modèle (model), perspective (perspective), photos (photos), rapport (report), exigences (requirements), spécifications (specifications), agenda, planning, to do
- Objet (object) : bâtiment (building), lot (plot), étage (level), pièce (room), élément de



construction (element), réservation (reservation), matériau (material), échantillon de matériau (material sample), véhicule (vehicle), défaut (defect)
- Message : Information, Réaction (Reaction), Notification, Requête (request), Validation
- Groupe d'artefacts (group of artifacts)

Les artefacts sont également définit par :
- Une description plus spécifique
- Un statut (status): en cours (ongoing), final, à valider (to validate), validé (valid), à exécuter (to execute), exécuté (executed), à modifier (to modify)
- Un auteur (author) : défini par son rôle (role)
-

Un acteur (actor) peut être responsable d'une pratique individuelle ou simplement impliqué (c'est à dire concerné par l'opération d'un autre acteur). Dans tout les cas il est définit par :




- Un rôle organisationnel (organizational role) : Concepteur (designer), Maître d'ouvrage (owner), Expert, Constructeur (constructor), Coordinateur (coordinator), Administration, Conseiller (advisor), autre (other)
- Un métier (job) : architecte (architect), ingénieur structure/ santé/sécurité (structure, safety, health engineer), urbaniste (urbaniste), Maçon (Mason), Electricien (Electrician), Charpentier (Carpenter), autre (other)…
- Une description plus spécifique

Les groupes d'acteur sont définit par les mêmes attributs et par un type de groupe.

Le projet (project) dans lequel se déroule la pratique collective est défini par :
- Un type : Habitat individuel (Individual Housing), Habitat collectif (Mass Housing), Bâtiment public (Public building)
- Une certification environnementale (certification type)

La phase de projet dans laquelle se déroule la pratique collective est définie par :
- Un type : Préparation, Conception (Design), Exécution, Livraison (Delivery)

La Tâche métier est définie par :
- Un type : de conception (design), d'exécution (execution), d'évaluation (assessment), de synthèse (synthesis), de coordination (coordination)
- Un auteur : défini par son rôle organisationnel

Le projet, les phases et les tâches sont considérées comme des activités et sont également définie par :
- des dates relatives : de début (strating date), de fin (ending date), d'interruption (interruption date), etc…
- Un type de durée : long (long) ou court (short)

*Chaque élément possède un champ d'édition libre destiné à en compléter la définition par toute information supplémentaire jugée utile à la description de la pratique.*

On retrouve plusieurs types de relations liant les différents éléments présentés ci-dessus :
Les relations de composition (is composed of): une Pratique Collective est composée de Pratiques Individuelles, une PI étant composée d'Opérations.
La relation exécute (executes) relie un acteur à une Pratique Individuelle.

La relation produit (creates) relie une opération à un artefact lorsque ce dernier est le fruit de l'opération qui le manipule (ex. créer un document).
La relation utilise (uses) relie une opération à un artefact lorsque ce dernier n'est pas le fruit de l'opération qui le manipule (ex. partagé un document créé)
La relation cible (targets) relie une opération à un acteur auquel elle a recours (ex. avertir le concepteur)

# Partie 2 : Usages

Comme introduit dans la description de l'analyse métier (voir 1.), il s'agit à présent de décrire la médiatisation des Pratiques métier Individuelles (PI) et donc des Opérations qui les composent à travers l'Usage d'un service informatique adapté.

-

La définition de l'usage se fait à travers :

- un diagramme de cas d'utilisation (use case diagram) dans lequel chaque opération (operation) sera décomposée en cas d'utilisations (use cases)
- une définition du contexte à travers des tableaux
- des modèles de tâches IHM détaillant chaque use case et si besoin des maquettages pour les illustrer
- une définition du contenu, à travers des tableaux décrivant l'information produite et/ou utilisée

-

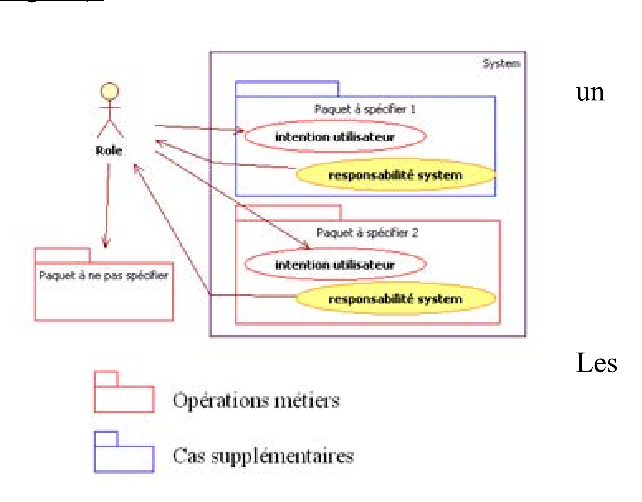
NB : Dans le cas ou une caractéristique ne peut être définie, cela sera précisé par une des 2 mentions suivantes :

- NIL : pas de réponse possible (ex. le système ne possède pas de type d'entrée vidéo)
- NC : réponse non communiquée (ex. le système possède peut-être un type d'entrée vidéo mais on ne sait pas laquelle).

***Légende et précisions :***

Le diagramme de cas d'utilisation (use case diagram):

L'acteur est définit par son rôle organisationnel. Chaque opération devient un paquet (package) de cas d'utilisation (use cases). Les paquets des opérations à spécifier sont représentés dans le système et décomposés. Le système peut aussi contenir des paquets de cas d'utilisation issus d'aucune opération métier mais néanmoins nécessaires et donc à définir. Les paquets des opérations non traitées sont quant à eux situés à l'extérieur du système.




Le contexte :

Il est définit sous forme de tableaux.

Le contexte général rappelle le nom de la Pratique Individuelle (Individual Practice) à outiller et résume le but de l'usage (ex. partager des documents à l'aide d'une plateforme d'échange/sharing documents with a groupware). Il définit également :

- La fréquence de l’usage (frequency) : rarement (rare), souvent (often), très souvent (very often)
- la localisation de l’usage (location) : au bureau (at office), sur le chantier (on site), au domicile (at home), chez le client (at customer), dans un lieu public (in public location)

Le contexte de l’acteur rappelle le métier de l’acteur et définit son rôle opérationnel (operationnal role), c.à.d. son rôle vis-à-vis du système (ex. administrateur, utilisateur…). Il détermine aussi si l’utilisateur doit être identifié.

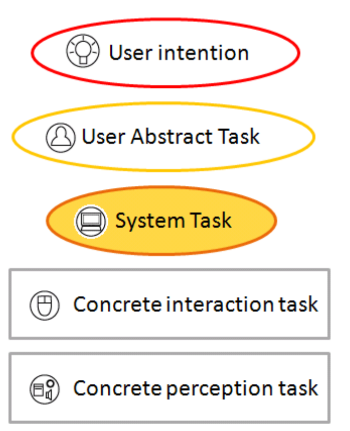
Le contexte logiciel (software) définit l’application à spécifier (type et description) et les technologies à utiliser pour cela.

Le contexte matériel (hardware) définit le support physique utilisé à savoir :

- le type d’appareil (device) : ordinateur personnel (computer), ordinateur portable (laptop), smartphone (smartphone), tablette tactile (tablet)…
- Le système d’exploitation (OS) : Windows, macOS, linux, android, iOS, WindowsPhone…
- Le type de dispositif d’interaction (interaction device) : clavier (keyboard) + souris (mouse), pavé tactile (touchpad), écran tactile (touchscreen)…
- La connexion à internet : connecté (online), déconnecté (offline)
- Le type de sortie vidéo (video output) : écran (screen), projecteur (projector)…
- Le type de sortie audio (audio output) : haut-parleurs (speakers), écouteurs (earspeakers)…
- Le type d’entrée vidéo (video input) : caméra (video camera), appareil photo numérique (photo camera)…
- Le type d’entrée audio (audio input) : microphone (microphone)

Chaque élément du contexte peut être précisé dans un deuxième tableau (additional information about context), le but étant d’apporter toute information qui pourra s’avérer utile au développement de la solution.

<u>Les tâches et le contenu d’interaction:</u>




Chaque intention utilisateur sera décomposée pour former un arbre de tâches comprenant plusieurs niveaux d’abstraction.
La tâche abtsraite décrit l’interaction de manière générale et indépendante de la technologie. Elle est définie par un nom et un type.
Deux types de tâches concrètes sont possibles : la tâche d’interaction (action physique de l’utilisateur sur le système) et la tâche de perception (retour visuel ou sonore du système pour informer l’utilisateur).
Les tâches système qui auront été identifiées dans le use case apparaissent également.

Chaque élément du contexte coopératif manipulé par une opération ainsi que défini dans la partie 1, devient un contenu à médiatiser lors de l’usage d’un service informatique, sous forme d’objet d’interaction. Ce contenu est alors identifié par un nom, une spécificité graphique, des attributs, un type de donnée et des propriétés.

Les relations entre tâches peuvent être de composition (is composed of) ou de processus (AND, OR, THEN). Une relation tâche-objet se nomme « interagit avec » (« interact with »).

|→| Then
|/| Or
|+| And

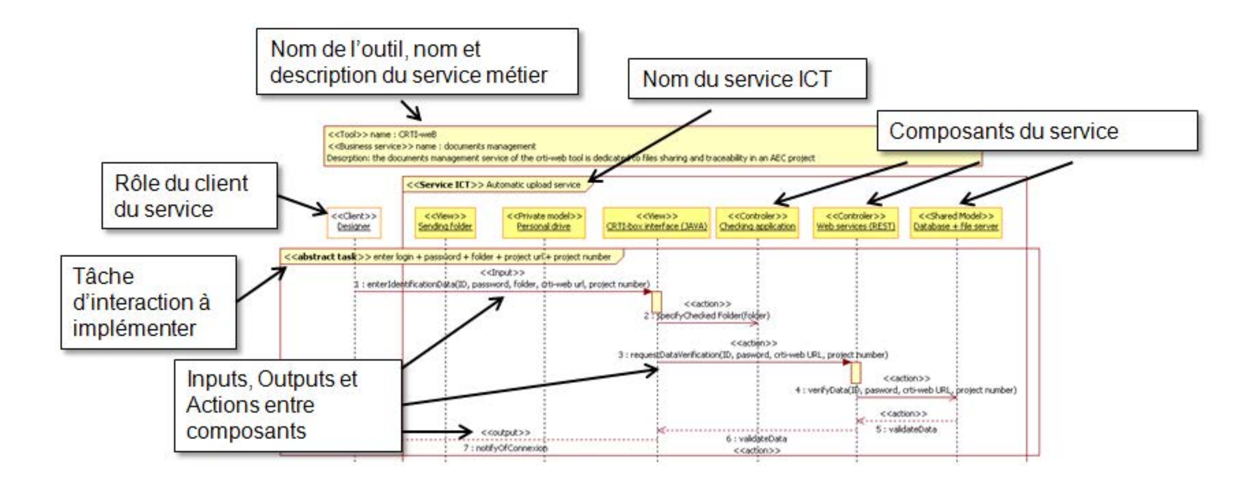
La description de l’interaction peut être complétée par des maquettages abstraits ou concrets

# Partie 3 : Spécification du service

La spécification du service apporte la réponse technologique aux usages définis dans l'étape précédente. Le formalisme utilisé est le diagramme de séquences.

***Légende et précisions :***

Les diagrammes de cette partie décrivent l'utilisation d'un service ICT par un client, l'architecture de ce service (chaque composant ayant un stéréotype « modèle privé », « modèle partagé », « vue » ou « contrôleur » et les flux de données (inputs du client vers la vue, outputs de la vue vers le client et actions entre les composants du service). Ces diagrammes prennent en compte des scénarios idéaux et les cas d'échecs possibles. Ils pourront évoluer en fonction de l'avancement du projet.



**Cahier d'exigences (partie éditable)**

# Requirements Specification Form for adapted ICT services design and development

*Requirement Form ID: Click here to enter text.*

## Name of the service to develop: Click here to enter text.

**Actors of the development project:**

Service designer: Click here to enter text.
Business analyst: Click here to enter text.
Developer: Click here to enter text.
Service consumer: Click here to enter text.

**Versioning:**

| Versions | Dates | Description of the project phase |
|---|---|---|
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |
| v x.x | Click here to enter a date. | Click here to enter text. |

## PART 1: BUSINESS PRACTICES

| PRACTICES DIAGRAM |
|---|
| Insert here the diagram illustrating the business practices. |
| ADDITIONAL INFORMATION |
| Do you want to add any information about the business context and the practice to perform?<br>Click here to enter text. |
| Do you want to add any information about the information to manage (artifacts, activities, actors)?<br>Click here to enter text. |
| |

# PART 2: USAGE

| USAGE DESCRIPTION | | | |
|---|---|---|---|
| Related Individual Practice | Click here to enter text. | | |
| Usage Objective | Click here to enter text. | | |
| USER INTENTIONS | | | |
| Insert here the use case diagram | | | |
| USAGE CONTEXT | | | |
| Temporality | Choose an item. | Location | Choose an item. |
| User | | Application | |
| Operational role | Click here to enter text. | Application type | Choose an item. |
| Identification | Choose an item. | Dev. technology | Click here to enter text. |
| Device | | | |
| Device type | Choose an item. | Video Output | Choose an item. |
| OS | Choose an item. | Audio Output | Choose an item. |
| Interaction device | Choose an item. | Video Input | Choose an item. |
| Connection | Choose an item. | Audio Input | Choose an item. |
| | | | |

| ADDITIONAL INFORMATION ABOUT CONTEXT |
| --- |
| Do you want to add any functional or non-functional requirements about… |
| The general context of the usage? |
| You can describe here the pre-conditions and post-conditions of the usage. It can refer to the relations with other usages.<br>Click here to enter text. |
| The actor's specificity as a user? |
| What do you know about the actor using the system? Eventually precise here, his skills, knowledge, abilities, preferences…<br>Click here to enter text. |
| The application specificity? |
| Is there already a developed application to adapt? You can describe it here.<br>Click here to enter text. |
| What about the new services to develop?<br>Click here to enter text. |
| The device specificity? |
| Does it have a particularity in terms of mobility, performance, autonomy, capacity…? Do you think about a specific model?<br>Click here to enter text. |
| |

| RELATED USAGES | | | |
|---|---|---|---|
| You can specify the link between the usage you described and others | | | |
| Form ID | Usage description | Relation in time | Relation in space |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| Click here to enter text. | Click here to enter text. | Choose an item. | Choose an item. |
| | | | |

## PART 3: SERVICE

| SERVICE SPECIFICATION |
|---|
| Insert here the sequence diagram |
| |

**Designing collaborative services adapted to business practices: a usage-centered method – Application to the construction sector**

## Abstract

In the sector of the architectural design-construction project, the management of the collaboration between the different actors of a project is an important issue. From a project to another, considering the project type and also the actors involved, business practices vary. In parallel, many services are proposed and used to assist the collaboration, some of them being generic and others more specific to professional usage. The CRTI-weB experience is a groupware development project lead with and for professionals of the Luxembourgish sector. It actually proposes two services in its commercial version: the documents exchange and management service (plans, technical documents…) and the site meeting report redaction and management service. Despite the analysis of business needs lead before development, the tool has some lacks of adaptation. This statement isn't isolated; in general, this type of tools doesn't fully satisfy professionals' expectations. This doctoral work proposes an analysis framework supported by users' behaviors modeling. We adopt several viewpoints related to several research fields: software engineering, human-computer interactions design, service-oriented enterprise and information system design, and finally computer-supported collaborative work (CSCW). From an organizational viewpoint, we identify collective practices performed by groups of actors involved in the project. Then, we deduce the individual practice, i.e. the responsibilities of each actor considering his role in the group. From an operational viewpoint, are interested in the mediation of these responsibilities by the usages of different technological solutions. The definition of these usages is linked to the concept of context: the technical context (hardware, software…), the place context (location, environment…) and the temporal context (frequency, regularity, synchronization…). Finally, the functional viewpoint aims at defining the business service, i.e. the behavior of the system that answers to one business need. We propose the PUSH method (Practice and Usages based Service enhancement) that orchestrates the different viewpoints to generate an amount of requirements for the development of adapted services. The communication and the traceability are supported by this design method. The context of study, both research and development-oriented (through the collaboration of the MAP-CRAI laboratory in Nancy and the CRP Henri Tudor in Luxembourg) allows us evaluating and enhancing the definition of our concepts and the applicability of the PUSH method through three experimentations.

**Keywords:** Construction, Business Practices, Collaborative services, Software Engineering, Human-Computer Interactions, Usage-centered design, Design science.

**RAMEAU Keywords:** Groupware, Software – Development, Collaborative work, User centered design, Design patterns, Architecture – Software engineering, Architecture -- Practices

**Concevoir des services collaboratifs adaptés à des pratiques métiers : une méthode centrée usages - Application au domaine de la construction**

## Résumé

Dans le domaine du projet de conception-construction architecturale, la gestion de la collaboration entre les différents acteurs d'un projet est un enjeu important. D'un projet à un autre, en fonction du projet, mais aussi des acteurs qui y interviennent, les pratiques de travail varient. Parallèlement, de nombreux services sont proposés et utilisés pour assister la collaboration, certains sur un modèle « grand public » et d'autres plus orientés vers un usage professionnel. L'expérience CRTI-weB est un projet de développement d'un collecticiel mené avec et pour les professionnels du secteur au Luxembourg. Il propose actuellement deux services dans sa version commerciale : le service d'échanges et gestion de documents (plans, documents techniques) et le service de rédaction et gestion des rapports de chantier. Malgré l'étude des besoins métiers menée en amont du développement, l'outil présente cependant des manques d'adaptation. Ce constat d'inadaptation n'est pas un cas isolé, il reflète une lacune générale de ce genre d'outils à satisfaire pleinement les attentes des professionnels. C'est pourquoi ce travail doctoral propose un cadre d'analyse supporté par la modélisation des comportements des utilisateurs. Nous adoptons alors plusieurs points de vue relatifs à différents champs de recherches : le génie logiciel, la conception d'interfaces homme-machine, l'entreprise orientée services et la conception de système d'information, et enfin le travail collaboratif assisté par ordinateur (TCAO). D'un point de vue organisationnel, nous identifions les pratiques collectives exercées par les groupes d'acteurs impliqués dans le projet. Nous en déduisons ensuite les pratiques individuelles, à savoir les responsabilités de chacun en fonction de son rôle dans le groupe. D'un point de vue opérationnel, nous nous intéressons à la médiatisation de ces responsabilités par l'usage de différentes solutions technologiques. La caractérisation de ces usages est liée à la notion de contexte : le contexte technique (matériel, logiciel…), le contexte spatial (localisation, environnement…), le contexte temporel (fréquence, régularité, synchronisation…). Enfin, le point de vue fonctionnel vise à définir le service utilisé, c'est-à-dire le comportement non plus de l'utilisateur, mais du système qui répond à son besoin. Nous proposons la méthode PUSH (Practices and Usages based Services enHancement) qui orchestre ces différents points de vue et permet de générer un ensemble d'exigences pour le développement de services dits « adaptés ». Communication et traçabilité sont les maitres mots de cette méthode de conception. Le contexte d'étude à la fois orienté recherche et développement nous a permis d'évaluer et d'améliorer la définition des concepts mis en avant ainsi que la mise en place de la méthode PUSH à travers trois expérimentations.

**Mots-clés :** Construction, Pratiques métiers, Services collaboratifs, Génie Logiciel, Interfaces Homme-Machines, Conception centrée usages, Science de la conception.

**Mots-clés RAMEAU :** Logiciels de groupe, Logiciels – Développement, Travail collaboratif, Conception centrée sur l'utilisateur, Design patterns, Architecture – Informatique, Architecture – Pratique.